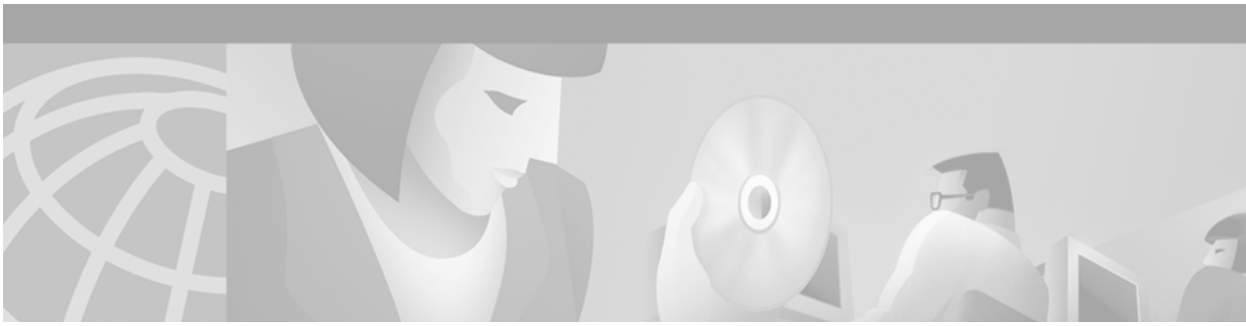

Configuring Novell IPX

This chapter describes how to configure Novell Internetwork Packet Exchange (IPX) and provides configuration examples. For a complete description of the IPX commands in this chapter, refer to the "Novell IPX Commands" chapter in the *Cisco IOS AppleTalk and Novell IPX Command Reference* publication. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

IPX Addresses

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

Network Numbers

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 hexadecimal digits. The maximum number of digits allowed is eight.

The Cisco IOS software does not require that you enter all eight digits; you can omit leading zeros.

Node Numbers

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

If you do not specify a node number for a router to be used on WAN links, the Cisco IOS software uses the hardware 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. If there are no valid IEEE interfaces, the Cisco IOS software randomly assigns a node number using a number that is based on the system clock.

IPX Address Example

The following example shows how to configure 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, perform the tasks in the following sections:

- Configuring IPX Routing (Required)
- Configuring IPX Enhanced IGRP (Optional)
- Configuring NLSP (Optional)
- Configuring Next Hop Resolution Protocol (Optional)
- Configuring IPX and SPX over WANs (Optional)
- Controlling Access to IPX Networks (Optional)
- Tuning IPX Network Performance (Optional)
- Shutting Down an IPX Network (Optional)
- Configuring IPX Accounting (Optional)
- Configuring IPX Between LANs (Optional)
- Configuring IPX Between VLANs (Optional)
- Configuring IPX Multilayer Switching (Optional)
- Monitoring and Maintaining the IPX Network (Optional)

See the "Novell IPX Configuration Examples" section in the Novell IPX Configuration Examples chapter.

Configuring IPX Routing

You configure IPX routing by first enabling it on the router and then configuring it on each interface.

Optionally, you can route IPX on some interfaces and transparently bridge it on other interfaces. You can also route IPX traffic between routed interfaces and bridge groups, or route IPX traffic between bridge groups.

To configure IPX routing, perform the tasks in the following sections. The first two tasks are required; the rest are optional.

- Enabling IPX Routing (Required)
- Assigning Network Numbers to Individual Interfaces (Required)
- Enabling Concurrent Routing and Bridging (Optional)
- Configuring Integrated Routing and Bridging (Optional)

IPX Default Routes

In IPX, a *default route* is the network where all packets for which the route to the destination address is unknown are forwarded.

Original Routing Information Protocol (RIP) implementations allowed the use of network -2 (0xFFFFFFFE) as a regular network number in a network. With the inception of NetWare Link Services Protocol (NLSP), network -2 is reserved as the default route for NLSP and RIP. Both NLSP and RIP routers should treat network -2 as a default route. Therefore, you should implement network -2 as the default route regardless of whether you configure NLSP in your IPX network.

By default, Cisco IOS software treats network -2 as the default route. You should ensure that your IPX network does not use network -2 as a regular network. If, for some reason, you must use network -2 as a regular network, you can disable the default behavior. To do so, see the "Adjusting Default Routes" section later in this chapter.

For more background information on how to handle IPX default routes, refer to the Novell *NetWare Link Services Protocol (NLSP) Specification, Revision 1.1* publication.

Enabling 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 to be used on WAN links, the Cisco IOS software uses the hardware 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. If there are no valid IEEE interfaces, the Cisco IOS software randomly assigns a node number using a number that is based on the system clock.

To enable IPX routing, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx routing [node]`	Enables IPX routing.

For an example of how to enable IPX routing, see the "IPX Routing Examples"section in the Novell IPX Configuration Examples 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.

Assigning Network Numbers to Individual Interfaces

After you have enabled IPX routing, you enable IPX routing on the individual interfaces by assigning network numbers to those interfaces.

You enable IPX routing on interfaces that support a single network or multiple networks.

When you enable IPX routing on an interface, you can also specify an encapsulation (frame type) to use for packets being sent on that network. Table 8 lists the encapsulation types you can use on IEEE interfaces and shows the correspondence between Cisco naming conventions and Novell naming conventions for the encapsulation types.

Table 8 Cisco and Novell IPX Encapsulation Names on IEEE Interfaces

Interface Type	Cisco Name	Novell Name
Ethernet	novell-ether (Cisco IOS default) arpa sap snap	Ethernet_802.3 Ethernet_II Ethernet_802.2 Ethernet_Snap
Token Ring	sap (Cisco IOS default) snap	Token-Ring Token-Ring_Snap
FDDI	snap (Cisco IOS default) sap novell-fddi	Fddi_Snap Fddi_802.2 Fddi_Raw

Note The SNAP encapsulation type is not supported and should not be configured on any IPX interfaces that are attached to a FDDI-Ethernet bridge.

Assigning Network Numbers to Individual Interfaces Task List

The following sections describe how to enable IPX routing on interfaces that support a single network and on those that support multiple networks. To enable IPX routing on an interface, you must perform one of the tasks:

- Assigning Network Numbers to Interfaces That Support a Single Network (Required)
- Assigning Network Numbers to Interfaces That Support Multiple Networks (Required)
- Setting the Encapsulation Type for Subinterfaces (Required)

Assigning Network Numbers to Interfaces That Support a Single 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.

To assign a network number to an interface that supports a single network, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx network` *network* `[encapsulation` *encapsulation-type*`]`	Enables IPX routing on an interface.

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. Novell-ether or ARPA encapsulations cannot be used for FDDI-Ethernet bridged IPX traffic. Use SAP encapsulations on originating and destination IPX interfaces that are attached to the FDDI-Ethernet bridge. See Table 8 for a list of encapsulation types you can use on IEEE interfaces.

For an example of how to enable IPX routing, see the ”IPX Routing Examples” section in the Novell IPX Configuration Examples chapter.

Assigning Network Numbers to Interfaces That Support Multiple Networks

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, the Cisco IOS software is allowed 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. Remember, the encapsulation type should match the servers and clients using the same network number. See Table 8 for a list of encapsulation types you can use on IEEE interfaces.

There are two ways to assign network numbers to interfaces that support multiple networks. You can use subinterfaces or primary and secondary networks.

Setting the Encapsulation Type for Subinterfaces

You typically use subinterfaces to assign network numbers to interfaces that support multiple networks.

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.

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

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, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number.subinterface-number`	Specifies a subinterface.
Step 2	`Router(config-if)# ipx network network [encapsulation encapsulation-type]`	Enables IPX routing, specifying the first encapsulation type.

Note You cannot configure more than 200 IPX interfaces on a router using the **ipx network** command.

To configure more than one subinterface, repeat these two steps. See Table 8 for a list of encapsulation types you can use on IEEE interfaces.

For examples of configuring multiple IPX networks on an interface, see the"IPX Routing on Multiple Networks Examples" section in the Novell IPX Configuration Examples chapter.

Primary and Secondary Networks

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

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, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ipx network` *network* `[encapsulation` *encapsulation-type*`]`	Enables IPX routing on the primary network.
Step 2	`Router(config-if)# ipx network` *network* `[encapsulation` *encapsulation-type*`] [secondary]`	Enables IPX routing on a secondary network.

To configure more than one secondary network, repeat these steps as appropriate. See Table 8 for a list of encapsulation types you can use on IEEE interfaces.

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

Enabling Concurrent Routing and Bridging

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

Command	Purpose
`Router(config)# bridge crb`	Enables the Cisco IOS software to both route and bridge a given protocol on separate interfaces within a single router.

Configuring Integrated Routing and Bridging

Integrated routing and bridging (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 *Cisco IOS Bridging and IBM Networking Configuration Guide*.

Configuring IPX Enhanced IGRP

Enhanced IGRP is an enhanced version of the Interior Gateway Routing Protocol (IGRP) developed by Cisco. 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 Features

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—Full update packets need not be processed each time they are received.
- Neighbor discovery mechanism—This feature is a simple hello mechanism used to learn about neighboring routers. It is protocol-independent.
- Scaling—Enhanced IGRP scales to large networks.

Enhanced IGRP Components

Enhanced IGRP has four basic components discussed in the following sections:

- Neighbor Discovery/Recovery
- Reliable Transport Protocol
- DUAL Finite-State Machine
- Protocol-Dependent Modules

Neighbor Discovery/Recovery

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. The router achieves neighbor discovery/recovery 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.

Reliable Transport Protocol

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 sent 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, which is indicated in the packet. The reliable transport has a provision to send multicast packets quickly when there are unacknowledged packets pending. This provision helps ensure that convergence time remains low in the presence of varying speed links.

DUAL Finite-State Machine

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.

Protocol-Dependent Modules

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.

IPX Enhanced IGRP Configuration Task List

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

- Enabling IPX Enhanced IGRP (Required)
- Customizing Link Characteristics (Optional)
- Customizing the Exchange of Routing and Service Information (Optional)
- Querying the Backup Server (Optional)

Enabling IPX Enhanced IGRP

To create an IPX Enhanced IGRP routing process, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router eigrp autonomous-system-number`	Enables an Enhanced IGRP routing process.
Step 2	`Router(config-if)# network {network-number | all}`	Enables Enhanced IGRP on a network.

To associate multiple networks with an Enhanced IGRP routing process, you can repeat the preceding two steps.

For an example of how to enable Enhanced IGRP, see the IPX Enhanced "IGRP Example" section in the Novell IPX Configuration Examples chapter.

Customizing Link Characteristics

You might want to customize the Enhanced IGRP link characteristics. The following sections describe these customization tasks:

- Configuring the Percentage of Link Bandwidth Used by Enhanced IGRP (Optional)
- Configuring Maximum Hop Count (Optional)
- Adjusting the Interval Between Hello Packets and the Hold Time (Optional)

Configuring 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 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, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx bandwidth-percent eigrp as-number percent`	Configures the percentage of bandwidth that may be used by Enhanced IGRP on an interface.

For an example of how to configure the percentage of Enhanced IGRP bandwidth, see the"IPX Enhanced IGRP Bandwidth Configuration Example"section in the Novell IPX Configuration Examples chapter.

Configuring Maximum Hop Count

Note Although adjusting the maximum hop count is possible, it is not recommended for Enhanced IGRP. We recommend that you use the default value for the maximum hop count of Enhanced IGRP.

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

Command	Purpose
`Router(config)# ipx maximum-hops hop`	Sets the maximum number of hops of an IPX packet reachable by non-RIP routing protocols. Also sets the maximum number of routers that an IPX packet can traverse before being dropped.

Adjusting 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 their neighbors 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 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 three times the hello interval, or 15 seconds.

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 increase the hold time, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx hold-time eigrp autonomous-system-number seconds`	Sets the hold time.

To change the interval between hello packets, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx hello-interval eigrp autonomous-system-number seconds`	Sets the interval between hello packets.

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

Customizing the Exchange of Routing and Service Information

You might want to customize the exchange of routing and service information. The following sections describe these customization tasks:

- Redistributing Routing Information (Optional)
- Disabling Split Horizon (Optional)
- Controlling the Advertising of Routes in Routing Updates (Optional)
- Controlling the Processing of Routing Updates (Optional)
- Controlling SAP Updates (Optional)
- Controlling the Advertising of Services in SAP Updates (Optional)
- Controlling the Processing of SAP Updates (Optional)

Redistributing Routing Information

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

To disable route redistribution, use the following command in IPX-router configuration mode:

Command	Purpose
`Router(config-ipx-router)# no redistribute {connected | eigrp autonomous-system-number | rip | static}`	Disables redistribution of RIP routes into Enhanced IGRP and Enhanced IGRP routes into RIP.

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

	Command	Purpose
Step 1	`Router(config)# ipx router eigrp autonomous-system-number`	From global configuration mode, enables Enhanced IGRP.
Step 2	`Router(config-ipx-router)# redistribute nlsp [tag]`	From IPX-router configuration mode, enables redistribution of NLSP into Enhanced IGRP.
Step 3	`Router(config)# ipx router nlsp [tag]`	Enables NLSP.
Step 4	`Router(config-ipx-router)# redistribute eigrp autonomous-system-number`	From IPX-router configuration mode, enables redistribution of Enhanced IGRP into NLSP.

For an example of how to enable redistribution of Enhanced IGRP and NLSP, see the "Enhanced IGRP and NLSP Route Redistribution Example" section in the Novell IPX Configuration Examples chapter.

Disabling 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, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no ipx split-horizon eigrp autonomous-system-number`	Disables split horizon.

Note Split horizon cannot be disabled for RIP or SAP, only for Enhanced IGRP.

Controlling 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 control this advertising, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# distribute-list access-list-number out [interface-name | routing-process]`	Controls the advertising of routes in routing updates.

Controlling the Processing of Routing Updates

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

Command	Purpose
`Router(config-router)# distribute-list access-list-number in [interface-name]`	Controls which incoming route updates are processed.

Controlling 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. This feature should only be disabled when all nodes out of 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, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx sap-incremental eigrp` *autonomous-system-number*	Sends SAP updates only when a change in the SAP table occurs.

To send SAP updates only when a change occurs in the SAP table and to send only the SAP changes, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx sap-incremental eigrp` *autonomous-system-number* **rsup-only**	Sends SAP updates only when a change in the SAP table occurs, and sends only the SAP changes.

When you enable incremental SAP using the **ipx sap-incremental eigrp rsup-only** command, Cisco IOS software disables the exchange of route information via Enhanced IGRP for that interface.

To send periodic SAP updates, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no ipx sap-incremental eigrp` *autonomous-system-number*	Sends SAP updates periodically.

For an example of how to configure SAP updates, see the"Enhanced IGRP SAP Update Examples" section in the Novell IPX Configuration Examples chapter.

To disable split horizon for incremental SAP, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no ipx sap-incremental split-horizon`	Disables split horizon for SAP.

Note IPX incremental SAP split horizon is off for WAN interfaces and subinterfaces, and on for LAN interfaces. The global default stays off. The interface setting takes precedence if the interface setting is modified or when both the global and interface settings are unmodified. The global setting is used only when the global setting is modified and the interface setting is unmodified.

Controlling 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 control this advertising, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# distribute-sap-list` *access-list-number* **out** [*interface-name* | *routing-process*]	Controls the advertising of services in SAP updates distributed between routing processes.

For a configuration example of controlling the advertisement of SAP updates, see the "Advertisement and Processing of SAP Update Examples" section in the Novell IPX Configuration Examples chapter.

Controlling the Processing of SAP Updates

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

Command	Purpose
`Router(config-router)# distribute-sap-list` *access-list-number* **in** [*interface-name*]	Controls which incoming SAP updates are processed.

For a configuration example of controlling the processing of SAP updates, see the "Advertisement and Processing of SAP Update Examples" section in the Novell IPX Configuration Examples chapter.

Querying 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 learn 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 need not be sent.

By default, the Cisco IOS software queries its own copy of the backup server table of each Enhanced IGRP neighbor every 60 seconds. To change this interval, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx backup-server-query-interval` *interval*	Specifies the minimum period of time between successive queries of the backup server table of a neighbor.

Configuring NLSP

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.

Understanding Level 1, 2, and 3 Routers

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 with area route aggregation and route redistribution has been defined in a specification.

Understanding NLSP Databases

NLSP is a link-state protocol, which means that every router in a routing area maintains an identical copy of the link-state database. This database 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 immediate neighbors of the router 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 2 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.

Cisco Support of NLSP

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

NLSP Configuration Task List

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:

- Defining an Internal Network (Required)
- Enabling NLSP Routing (Required)
- Configuring NLSP on an Interface (Required)

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

- Customizing Link Characteristics (Optional)
- Configuring Route Aggregation (Optional)
- Customizing the Exchange of Routing Information (Optional)

For an example of enabling NLSP, see the "IPX Routing Protocols Examples" section in the Novell IPX Configuration Examples chapter.

Defining 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, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx routing`	Enables IPX routing.
Step 2	`Router(config)# ipx internal-network network-number`	Defines an internal network number.

Enabling NLSP Routing

To enable NLSP, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router nlsp [tag]`	Enables NLSP.
Step 2	`Router(config-if)# area-address address mask`	Defines a set of network numbers to be part of the current NLSP area.

Configuring NLSP on an Interface

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

- Configuring NLSP on a LAN Interface (Required)
- Configuring NLSP on a WAN Interface (Required)

Configuring NLSP on a LAN Interface

To configure NLSP on a LAN interface, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ipx network network [encapsulation encapsulation-type]`	Enables IPX routing on an interface.
Step 2	`Router(config-if)# ipx nlsp [tag] enable`	Enables NLSP on the interface.

To configure multiple encapsulations on the same physical LAN interfaces, you must configure subinterfaces. Each subinterface must have a different encapsulation type. To configure subinterfaces, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number.subinterface-number`	Specifies a subinterface.
Step 2	`Router(config-if)# ipx network network [encapsulation encapsulation-type]`	Enables IPX routing, specifying the first encapsulation type.
Step 3	`Router(config-if)# ipx nlsp [tag] enable`	Enables NLSP on the subinterface.

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.

Configuring NLSP on a WAN Interface

To configure NLSP on a WAN interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial number`	Specifies a serial interface.
Step 2	`Router(config-if)# ipx ipxwan [local-node unnumbered local-server-name retry-interval retry-limit]`	Enables IPXWAN.
Step 3	`Router(config-if)# ipx nlsp [tag] enable`	Enables NLSP on the interface.

Customizing Link Characteristics

You might want to customize the NLSP link characteristics. The following sections describe these customization tasks:

- Enabling NLSP Multicast Addressing (Optional)
- Configuring the Metric Value (Optional)
- Configuring the Link Delay and Throughput (Optional)
- Configuring the Maximum Hop Count (Optional)
- Specifying a Designated Router (Optional)
- Configuring Transmission and Retransmission Intervals (Optional)
- Modifying LSP Parameters (Optional)
- Limiting Partial Route Calculations (Optional)

Enabling NLSP Multicast Addressing

Cisco IOS supports the use of NLSP multicast addressing for Ethernet, Token Ring, and FDDI router interfaces. This capability is only possible when the underlying Cisco hardware device or driver supports multicast addressing.

With this feature, the router defaults to using multicasts on Ethernet, Token Ring, and FDDI interfaces, instead of broadcasts, to address all NLSP routers on the network. If an adjacent neighbor does not support NLSP multicasting, the router will revert to using broadcasts on the affected interface.

This feature is only available on routers running Cisco IOS software Release 11.3 or later. When routers running prior versions of Cisco IOS software are present on the same network with routers running Cisco IOS Release 11.3 software, broadcasts will be used on any segment shared by the two routers.

The NLSP multicast addressing offers the following benefits:

- Increases overall efficiency and performance by reducing broadcast traffic
- Reduces CPU cycles on devices that use NLSP multicast addressing
- Increases the Cisco level of compliance with the Novell NLSP specification, version 1.1

NLSP Multicast Addressing

By default, NLSP multicast addressing is enabled. You need not configure anything to turn on NLSP multicasting.

Typically, you do not want to substitute broadcast addressing where NLSP multicast addressing is available. NLSP multicast addressing uses network bandwidth more efficiently than broadcast addressing. However, there are circumstances where you might want to disable NLSP multicast addressing.

For example, you might want to disable NLSP multicast addressing in favor of broadcast addressing when one or more devices on a segment do not support NLSP multicast addressing. You might also want to disable it for testing purposes.

If you want to disable NLSP multicast addressing, you can do so for the entire router or for a particular interface.

To disable multicast addressing for the entire router, use the following commands in IPX-router configuration mode:

	Command	Purpose
Step 1	`Router(config-ipx-router)# ipx router nlsp`	Enters NLSP router configuration mode.
Step 2	`Router(config-ipx-router)# no multicast`	Disables NLSP multicast addressing on the router.

To disable multicast addressing on a particular router interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no ipx nlsp [tag] multicast`	Disables multicast addressing on the interface.

For examples of how to disable NLSP multicast addressing, see the "NLSP Multicast Addressing Examples" section in the Novell IPX Configuration Examples chapter.

Configuring the Metric Value

NLSP assigns a default link cost (metric) based on the link throughput. If desired, you can set the link cost manually.

Typically, you need not set the link cost manually; however, there are some cases where you might want to. For example, in highly redundant networks, you might want to favor one route over another for certain kinds of traffic. As another example, you might want to ensure load sharing. Changing the metric value can help achieve these design goals.

To set the NLSP link cost for an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx nlsp [tag] metric metric-number`	Sets the metric value for an interface.

Configuring 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.

Typically, you need not change the link delay and throughput; however, there are some cases where you might want to change these parameters. For example, in highly redundant networks, you might want to favor one route over another for certain kinds of traffic. To favor one route over another, you would change the metric on the less-desirable path to be slightly worse by assigning it a higher metric value using the **ipx-link-delay** command. In this case, traffic is forced to route over the favorable path. As another example, you might want to ensure load sharing. To load share, you would ensure that the metrics on the equal paths are the same.

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, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx link-delay microseconds`	Specifies the link delay.

To change the throughput, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx throughput bits-per-second`	Specifies the throughput.

Configuring the Maximum Hop Count

By default, IPX packets whose hop count exceeds 15 are discarded. In larger internetworks, this maximum hop count may be insufficient. You can increase the hop count to a maximum of 127 hops for NLSP.

For example, if you have a network with end nodes separated by more than 15 hops, you can set the maximum number of hops considered to be reachable by non-RIP routing protocols to a value from 16 to 127.

To modify the maximum hop count, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx maximum-hops hop`	Sets the maximum number of hops of an IPX packet reachable by non-RIP routing protocols. Also sets the maximum number of routers that an IPX packet can traverse before being dropped.

Specifying a Designated Router

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 sends it to the remainder of the routing area. The routing information generated includes adjacencies and RIP routes. The use of a designated router substantially 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 this priority, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx nlsp [tag] priority priority-number`	Configures the designated router election priority.

Configuring 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, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx nlsp [tag] 3.325 seconds`	Configures the hello transmission interval.

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

Command	Purpose
`Router(config-if)# ipx nlsp [tag] hello-multiplier multiplier`	Configures the hello multiplier.

Although not typically necessary, you can configure the CSNP transmission interval. To configure the CSNP interval, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx nlsp [tag] csnp-interval seconds`	Configures the CSNP transmission interval.

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

Command	Purpose
`Router(config-if)# ipx nlsp [tag] lsp-interval interval`	Configures the LSP transmission interval.

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

Command	Purpose
`Router(config-if)# ipx nlsp [tag] retransmit-interval seconds`	Configures the LSP retransmission interval.

Modifying LSP Parameters

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

Command	Purpose
`Router(config-router)# lsp-gen-interval seconds`	Sets the minimum LSP generation interval.
`Router(config-router)# max-lsp-lifetime [hours] value`	Sets the maximum time the LSP persists.
`Router(config-router)# lsp-refresh-interval seconds`	Sets the LSP refresh time.
`Router(config-router)# lsp-mtu bytes`	Sets the maximum size of an LSP.
`Router(config-router)# spf-interval seconds`	Sets the minimum time between SPF calculations.

Limiting Partial Route Calculations

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

To modify the PRC, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# prc-interval seconds`	Sets the hold-down period between partial route calculations.

Configuring 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 or later software, you can easily perform 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 the Novell *NetWare Link Services Protocol (NLSP) Specification, Revision 1.1* publication.

Note In the sections that follow, "NLSP version 1.1 routers" refer to routers that support the route aggregation feature, while "NLSP version 1.0 routers" refer 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.

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 FFFFFF00 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 FFFFFF00 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 FFFFFF00 is 18 12345600.

Relationship Between Area Addresses and Route Summaries

When you enable route summarization in Cisco IOS 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 behavior of the router 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, the link-state database of each router 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 default route redistribution behavior of the router 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. See the "Redistributing Routing Information" section later 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.

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 need not be propagated to neighboring areas.
- Reduced computational overhead—Because 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 of these benefits, you can build larger IPX networks using route aggregation.

Default Route Summarization

When you explicitly enable route summarization, the default route summarization depends on 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 later in this chapter.
- 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 later in this chapter.
- 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 later in this chapter.

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 route summarization behavior of the router 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. You can use numbered or named access lists 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 "Novell IPX Commands" chapter in the *Cisco IOS AppleTalk and Novell IPX Command Reference* publication.

Filtering is independent of route summarization, but may affect it indirectly, because 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 network of the service is reachable by an explicit route, an aggregated route, or the default route. When a server for a Get Nearest Server (GNS) response is chosen, 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.

Route Aggregation Configuration Task List

To configure the route aggregation feature, perform one or more of the task in the following sections:

- Configuring Route Aggregation for Multiple NLSP Version 1.1 Areas (Optional)
- Configuring Route Aggregation for NLSP Version 1.1 and NLSP Version 1.0 Areas (Optional)
- Configuring Route Aggregation for Enhanced IGRP and NLSP Version 1.1 Environments (Optional)
- Configuring Route Aggregation for RIP and NLSP Version 1.1 Environments (Optional)

Configuring 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. 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.

The following sections describe how to configure route aggregation for multiple NLSP version 1.1 areas:

- Configuring Route Aggregation with Default Route Summarization
- Configuring Route Aggregation with Customized Route Summarization Using Numbered Access Lists
- Configuring Route Aggregation with Customized Route Summarization Using Named Access Lists

Configuring Route Aggregation with Default Route Summarization

To configure the route aggregation feature with the default route summarization behavior, use the following commands beginning in global configuration mode for each NLSP process:

	Command	Purpose
Step 1	`Router(config)# ipx router nlsp [tag]`	Enables NLSP routing and identifies the process with a unique tag.
Step 2	`Router(config-router)# area-address address mask`	From router configuration mode, defines up to three area addresses for the process.
Step 3	`Router(config-router)# route-aggregation`	(Optional) From router configuration mode, enables route summarization.
Step 4	`Router(config-if)# ipx nlsp [tag] enable`	From interface configuration mode, enables NLSP on each network in the area described by the *tag* argument.

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 in the Novell IPX Configuration Examples chapter.

Configuring Route Aggregation with Customized Route Summarization Using Numbered Access Lists

To configure the route aggregation feature with customized route summarization behavior (using numbered access lists), use the following commands beginning in global configuration mode for each NLSP process:

	Command	Purpose
Step 1	`Router(config)# ipx router nlsp [tag]`	Enables NLSP routing and identifies the process with a unique tag.
Step 2	`Router(config-router)# area-address address mask`	From router configuration mode, defines up to three area addresses for the process.
Step 3	`Router(config-router)# route-aggregation`	From router configuration mode, enables route summarization.
Step 4	`Router(config-router)# redistribute nlsp [tag] access-list access-list-number`	From router configuration mode, use the **redistribute** command with an access list in the range of 1200 to 1299. In this case, the *tag* argument identifies a unique NLSP process.
Step 5	`Router(config-if)# ipx nlsp [tag] enable`	From interface configuration mode, enables NLSP on each network in the area described by the *tag* argument.
Step 6	`Router(config)# access-list access-list-number deny network network-mask [ticks ticks] [area-count area-count]`	From global configuration mode, defines 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.
Step 7	`Router(config)# access-list access-list-number permit -1`	(Optional) Terminates the access list with a "permit all" statement to redistribute all other routes as explicit routes.

Configuring Route Aggregation with Customized Route Summarization Using Named Access Lists

To configure the route aggregation feature with customized route summarization behavior (using named access lists), use the following commands beginning in global configuration mode for each NLSP process:

	Command	Purpose
Step 1	`Router(config)# ipx router nlsp [tag]`	Enables NLSP routing and identifies the process with a unique tag.
Step 2	`Router(config-router)# area-address address mask`	From router configuration mode, defines up to three area addresses for the process.
Step 3	`Router(config-router)# route-aggregation`	From router configuration mode, enables route summarization.

	Command	Purpose
Step 4	`Router(config-router)# redistribute nlsp [tag] access-list name`	From router configuration mode, redistributes NLSP version 1.0 into the NLSP version 1.1 area. In this case, a named access list is used and the *tag* argument identifies a unique NLSP process.
Step 5	`Router(config-if)# ipx nlsp [tag] enable`	From interface configuration mode, enables NLSP on each network in the area described by the *tag* argument.
Step 6	`Router(config)# ipx access-list summary name`	From global configuration mode, specifies a named IPX access list for NLSP route aggregation.
Step 7	`Router(config-access-list)# deny network network-mask [ticks ticks] [area-count area-count]`	In access-list configuration mode, specifies the redistribution of aggregated routes instead of explicit routes. For each address range you want to summarize, use a **deny** command.
Step 8	`Router(config-access-list)# permit -1`	(Optional) Terminates the access list with a "permit all" statement to redistribute all other routes as explicit routes.

Configuring 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.

The following sections describe how to configure route aggregation for NLSP version 1.1 and NLSP version 1.0 areas:

- Configuring Route Aggregation with Default Route Summarization
- Configuring Route Aggregation with Customized Route Summarization Using Numbered Access Lists
- Configuring Route Aggregation with Customized Route Summarization Using Named Access Lists

Configuring Route Aggregation with Default Route Summarization

To configure the route aggregation feature with default route summarization behavior, use the following commands beginning in global configuration mode for each NLSP process:

	Command	Purpose
Step 1	`Router(config)# ipx router nlsp [tag]`	Enables NLSP routing and identifies the process with a unique tag.
Step 2	`Router(config-router)# area-address address mask`	From router configuration mode, defines up to three area addresses for the process.
Step 3	`Router(config-router)# route-aggregation`	For NLSP version 1.1 areas, enables route summarization from router configuration mode. Omit this step for NLSP version 1.0 areas.
Step 4	`Router(config-if)# ipx nlsp [tag] enable`	From interface configuration mode, enables NLSP on each network in the area described by the *tag* argument.

Configuring Route Aggregation with Customized Route Summarization Using Numbered Access Lists

To configure the route aggregation feature with customized route summarization behavior (using numbered access lists), use the commands in the following two tables.

For the NLSP version 1.1 process, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router nlsp [tag]`	Enables NLSP routing and identifies the process with a unique tag.
Step 2	`Router(config-router)# area-address address mask`	From router configuration mode, defines up to three area addresses for the process.
Step 3	`Router(config-router)# route-aggregation`	From router configuration mode, enables route summarization for NLSP version 1.1 areas.
Step 4	`Router(config-router)# redistribute nlsp [tag] access-list access-list-number`	(Optional) From router configuration mode, redistributes NLSP version 1.0 into the NLSP version 1.1 area. Include an access list number from 1200 to 1299.
Step 5	`Router(config-if)# ipx nlsp [tag] enable`	From interface configuration mode, enables NLSP on each network in the area described by the *tag* argument.
Step 6	`Router(config)# access-list access-list-number deny network network-mask [ticks ticks] [area-count area-count]`	(Optional) From global configuration mode, defines 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.
Step 7	`Router(config)# access-list access-list-number permit -1`	(Optional) From global configuration mode, terminates the access list with a "permit all" statement to redistribute all other routes as explicit routes.

For the NLSP version 1.0 process, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router nlsp [tag]`	Enables NLSP routing and identifies the process with a unique tag.
Step 2	`Router(config-router)# area-address address mask`	From router configuration mode, defines up to three area addresses for the process.
Step 3	`Router(config-if)# ipx nlsp [tag] enable`	From interface configuration mode, enables NLSP on each network in the area described by the *tag* argument.

For an example of how to configure the route aggregation feature with this type of customized route summarization, see the "NLSP Route Aggregation for NLSP Version 1.1 and Version 1.0 Areas Example"section in the Novell IPX Configuration Examples chapter.

Configuring Route Aggregation with Customized Route Summarization Using Named Access Lists

To configure the route aggregation feature with customized route summarization behavior (using named access lists), use the commands in the following two tables.

For the NLSP version 1.1 process, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router nlsp [tag]`	Enables NLSP routing and identifies the process with a unique tag.
Step 2	`Router(config-router)# area-address address mask`	From router configuration mode, defines up to three area addresses for the process.
Step 3	`Router(config-router)# route-aggregation`	From router configuration mode, enables route summarization for NLSP version 1.1 areas.
Step 4	`Router(config-router)# redistribute nlsp [tag] access-list name`	(Optional) From router configuration mode, redistributes NLSP version 1.0 into the NLSP version 1.1 area.
Step 5	`Router(config-if)# ipx nlsp [tag] enable`	From interface configuration mode, enables NLSP on each network in the area described by the *tag* argument.
Step 6	`Router(config)# ipx access-list summary name`	(Optional) From global configuration mode, specifies a named IPX access list for NLSP route aggregation.
Step 7	`Router(config-access-list)# deny network network-mask [ticks ticks] [area-count area-count]`	(Optional) From access-list configuration mode, defines 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 a **deny** statement.
Step 8	`Router(config-access-list)# permit -1`	(Optional) From access-list configuration mode, terminates the access list with a "permit all" statement to redistribute all other routes as explicit routes.

For the NLSP version 1.0 process, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router nlsp [tag]`	Enables NLSP routing and identifies the process with a unique tag.
Step 2	`Router(config-router)# area-address address mask`	From router configuration mode, defines up to three area addresses for the process.
Step 3	`Router(config-if)# ipx nlsp [tag] enable`	From interface configuration mode, enables NLSP on each network in the area described by the *tag* argument.

Configuring 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.

The following sections describe how to configure route aggregation for Enhanced IGRP and NLSP version 1.1 environments:

- Configuring Route Aggregation Using Numbered Access Lists
- Configuring Route Aggregation Using Named Access Lists

Configuring Route Aggregation Using Numbered Access Lists

For each NLSP version 1.1 process, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router nlsp [tag]`	Enables NLSP routing and identifies the process with a unique tag.
Step 2	`Router(config-router)# area-address address mask`	From router configuration mode, defines up to three area addresses for the process.
Step 3	`Router(config-router)# route-aggregation`	(Optional) From router configuration mode, enables route summarization.
Step 4	`Router(config-router)# redistribute {eigrp autonomous-system-number} [access-list access-list-number]`	(Optional) From router configuration mode, redistributes Enhanced IGRP into the NLSP version 1.1 area. Include an access list number from 1200 to 1299.
Step 5	`Router(config-if)# ipx nlsp [tag] enable`	From interface configuration mode, enables NLSP on each network in the area described by the *tag* argument.
Step 6	`Router(config)# access-list access-list-number deny network network-mask [ticks ticks] [area-count area-count]`	(Optional) From global configuration mode, defines 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.
Step 7	`Router(config)# access-list access-list-number permit -1`	(Optional) Terminates the access list with a "permit all" statement to redistribute all other Enhanced IGRP routes as explicit routes.

For each Enhanced IGRP autonomous system, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router eigrp autonomous-system-number`	Enables Enhanced IGRP.
Step 2	`Router(config-router)# network {network-number | all}`	From router configuration mode, specifies the networks to be enabled for Enhanced IGRP.
Step 3	`Router(config-router)# redistribute nlsp [tag]`	From router configuration mode, redistributes NLSP version 1.1 into Enhanced IGRP.

For an example of how to configure this type of route aggregation, see the "NLSP Route Aggregation for NLSP Version 1.1, Enhanced IGRP, and RIP Example"section in the Novell IPX Configuration Examples chapter.

Configuring Route Aggregation Using Named Access Lists

For each NLSP version 1.1 process, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router nlsp [tag]`	Enables NLSP routing and identifies the process with a unique tag.
Step 2	`Router(config-router)# area-address address mask`	From router configuration mode, defines up to three area addresses for the process.
Step 3	`Router(config-router)# route-aggregation`	(Optional) From router configuration mode, enables route summarization.
Step 4	`Router(config-router)# redistribute {eigrp autonomous-system-number} access-list name`	(Optional) From router configuration mode, redistributes Enhanced IGRP into the NLSP version 1.1 area.
Step 5	`Router(config-if)# ipx nlsp [tag] enable`	From interface configuration mode, enables NLSP on each network in the area described by the *tag* argument.
Step 6	`Router(config)# ipx access-list summary name`	(Optional) From global configuration mode, specifies a named IPX access list for NLSP route aggregation.
Step 7	`Router(config-access-list)# deny network network-mask [ticks ticks] [area-count area-count]`	(Optional) From access-list configuration mode, defines 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 a **deny** statement.
Step 8	`Router(config)# permit -1`	(Optional) From global configuration mode, terminates the access list with a "permit all" statement to redistribute all other Enhanced IGRP routes as explicit routes.

For each Enhanced IGRP autonomous system, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router eigrp autonomous-system-number`	Enables Enhanced IGRP.
Step 2	`Router(config-router)# network {network-number | all}`	From router configuration mode, specifies the networks to be enabled for Enhanced IGRP.
Step 3	`Router(config-router)# redistribute nlsp [tag]`	From router configuration mode, redistributes NLSP version 1.1 into Enhanced IGRP.

Configuring 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.

The following sections describe how to configure route aggregation for RIP and NLSP version 1.1 environments:

- Configuring Route Aggregation Using Numbered Access Lists
- Configuring Route Aggregation Using Named Access Lists

For an example of how to configure this type of route aggregation, see the "NLSP Route Aggregation for NLSP Version 1.1, Enhanced IGRP, and RIP Example"section in the Novell IPX Configuration Examples chapter.

Configuring Route Aggregation Using Numbered Access Lists

For each NLSP version 1.1 process, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router nlsp [tag]`	Enables NLSP routing and identifies the process with a unique tag.
Step 2	`Router(config-router)# area-address address mask`	From router configuration mode, defines up to three area addresses for the process.
Step 3	`Router(config-router)# route-aggregation`	(Optional) From router configuration mode, enables route summarization.
Step 4	`Router(config-router)# redistribute rip [access-list access-list-number]`	(Optional) From router configuration mode, redistributes RIP routes into the NLSP version 1.1 area. Include an access list number from 1200 to 1299.
Step 5	`Router(config-if)# ipx nlsp [tag] enable`	From interface configuration mode, enables NLSP on each network in the area described by the *tag* argument.
Step 6	`Router(config)# access-list access-list-number deny network network-mask [ticks ticks] [area-count area-count]`	(Optional) From global configuration mode, defines 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	`Router(config)# access-list access-list-number permit -1`	(Optional) From global configuration mode, terminates the access list with a "permit all" statement to redistributes all other RIP routes as explicit routes.

For an example of how to configure this type of route aggregation, see the "NLSP Route Aggregation for NLSP Version 1.1, Enhanced IGRP, and RIP Example" section in the Novell IPX Configuration Examples chapter.

Configuring Route Aggregation Using Named Access Lists

For each NLSP version 1.1 process, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router nlsp [tag]`	Enables NLSP routing and identifies the process with a unique tag.
Step 2	`Router(config-router)# area-address address mask`	From router configuration mode, defines up to three area addresses for the process.
Step 3	`Router(config-router)# route-aggregation`	(Optional) From router configuration mode, enables route summarization.
Step 4	`Router(config-router)# redistribute rip access-list name`	(Optional) From router configuration mode, redistributes RIP routes into the NLSP version 1.1 area.
Step 5	`Router(config-if)# ipx nlsp [tag] enable`	From interface configuration mode, enables NLSP on each network in the area described by the *tag* argument.
Step 6	`Router(config)# ipx access-list summary name`	(Optional) From global configuration mode, specifies a named IPX access list for NLSP route aggregation.
Step 7	`Router(config-access-list)# deny network network-mask [ticks ticks] [area-count area-count]`	(Optional) From access-list configuration mode, defines the access list to redistribute an aggregated route instead of explicit RIP routes. For each address range you want to summarize, use a **deny** statement.
Step 8	`Router(config-access-list)# permit -1`	(Optional) From access-list configuration mode, terminates the access list with a "permit all" statement to redistribute all other RIP routes as explicit routes.

Customizing the Exchange of Routing Information

You might want to customize the exchange of routing information. The following sections describe customization tasks:

- Configuring RIP and SAP Compatibility (Optional)
- Redistributing Routing Information (Optional)

Configuring RIP and SAP Compatibility

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, use one of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx nlsp [tag] rip off`	Never generates RIP periodic traffic.
`Router(config-if)# ipx nlsp [tag] rip on`	Always generates RIP periodic traffic.
`Router(config-if)# ipx nlsp [tag] rip auto`	Sends RIP periodic traffic only if another RIP router is sending periodic RIP traffic. (This is the default on interfaces configured for NLSP.)

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

Command	Purpose
`Router(config-if)# ipx nlsp [tag] sap off`	Never generates SAP periodic traffic.
`Router(config-if)# ipx nlsp [tag] sap on`	Always generates SAP periodic traffic.
`Router(config-if)# ipx nlsp [tag] sap auto`	Sends SAP periodic traffic only if another SAP service is sending periodic SAP traffic. (This is the default on interfaces configured for NLSP.)

Redistributing 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, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router nlsp [tag]`	Enables NLSP.
Step 2	`Router(config-ipx-router)# redistribute eigrp autonomous-system-number`	From IPX-router configuration mode, enables redistribution of Enhanced IGRP into NLSP.
Step 3	`Router(config)# ipx router eigrp autonomous-system-number`	From global configuration mode, enables Enhanced IGRP.
Step 4	`Router(config-ipx-router)# redistribute nlsp [tag]`	From IPX-router configuration mode, enables redistribution of NLSP into Enhanced IGRP.

For an example of how to enable redistribution of Enhanced IGRP and NLSP, see the “Enhanced IGRP and NLSP Route Redistribution Example”section in the Novell IPX Configuration Examples chapter.

Configuring Next Hop Resolution Protocol

Routers, access servers, and hosts can use Next Hop Resolution Protocol (NHRP) to discover the addresses of other routers and hosts connected to an NBMA network. NHRP provides an Address Resolution Protocol (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 the Cisco implementation, refer to the "Configuring IP Addressing" chapter in the *Cisco IOS IP Routing Configuration Guide.*

NHRP Configuration Task List

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

- Enabling NHRP on an Interface (Required)
- Configuring a Station with Static IPX-to-NBMA Address Mapping (Optional)
- Statically Configuring a Next Hop Server (Optional)
- Configuring NHRP Authentication (Optional)
- Controlling NHRP Initiation (Optional)
- Controlling NHRP Packet Rate (Optional)
- Suppressing Forward and Reverse Record Options (Optional)
- Specifying the NHRP Responder Address (Optional)
- Changing the Time Period NBMA Addresses Are Advertised As Valid (Optional)

For NHRP configuration examples, see the "NHRP Examples"section in the Novell IPX Configuration Examples chapter.

Enabling NHRP on an Interface

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

Command	Purpose
`Router(config-if)# ipx nhrp network-id` *number*	Enables NHRP on an interface.

For an example of enabling NHRP, see the "NHRP Examples"section in the Novell IPX Configuration Examples chapter.

Configuring a Station with 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 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 default or peer routers of the station, so their IPX addresses are obtained from the network layer forwarding table of the station.

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), use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx nhrp map` *ipx-address nbma-address*	Configures static IPX-to-NBMA address mapping.

Statically Configuring 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, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx nhrp nhs` *nhs-address* [*net-address*]	Statically configures a Next Hop Server.

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.

Configuring NHRP Authentication

Configuring an authentication string ensures that only routers configured with the same string can communicate 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, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx nhrp authentication` *string*	Specifies an authentication string.

Controlling NHRP Initiation

To control when NHRP is initiated, perform one of the tasks in the following sections:

- Triggering NHRP by IPX Packet (Optional)
- Triggering NHRP on a per-Destination Basis (Optional)

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, use one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# access-list access-list-number {deny | permit} source-network [.source-node [source-node-mask]][destination.network [.destination-node [destination-node-mask]]]`	Defines a standard IPX access list.
`Router(config)# access-list access-list-number {deny | permit} protocol [source-network] [[[.source-node] source-node-mask] | [.source-node source-network-mask.source-node-mask]] [source-socket] [destination.network] [[[.destination-node] destination-node-mask] | [.destination-node destination-network-mask .destination-node-mask]] [destination-socket]`	Defines an extended IPX access list.

To apply the IPX access list to the interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx nhrp interest access-list-number`	Specifies an IPX access list that controls NHRP requests.

Triggering NHRP on a per-Destination Basis

By default, when the software attempts to send a data packet to a destination for which it has determined that NHRP can be used, it sends 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 configure the system in this way, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx nhrp use usage-count`	Specifies how many data packets are sent to a destination before NHRP is attempted.

Controlling 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 sent. To change this maximum rate, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx nhrp max-send pkt-count every interval`	Changes the NHRP packet rate per interface.

Suppressing 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, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no ipx nhrp record`	Suppresses forward and reverse record options.

Specifying 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, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx nhrp responder type number`	Specifies which interface the Next Hop Server uses to determine the NHRP responder address.

If an NHRP reply packet being forwarded by a Next Hop Server contains the IPX address of that Next Hop Server, the Next Hop Server generates an “NHRP Loop Detected” error indication and discards the reply.

Changing the Time Period NBMA Addresses Are Advertised As Valid

You can change the length of time for which 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 7200 seconds (2 hours). To change the length of time, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx nhrp holdtime` *seconds-positive* [*seconds-negative*]	Specifies the number of seconds for which NBMA addresses are advertised as valid in positive or negative NHRP responses.

Configuring IPX and SPX over WANs

You can configure IPX over dial-on-demand routing (DDR), Frame Relay, PPP, SMDS, and X.25 networks. For more information about dial-on-demand routing (DDR) refer to the *Cisco IOS Dial Technologies Configuration Guide.* For more information about Frame Relay, SMDS, and X.25 refer to the *Cisco IOS Wide-Area Networking Configuration Guide*.

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 Subnetwork Access Protocol (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 in the Novell IPX Configuration Examples chapter.

Configuring IPX over DDR

IPX sends periodic watchdog keepalive packets from servers to clients after a client session has been idle for approximately 5 minutes. On a DDR link, 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 watchdog keepalive packets of a server on behalf of a remote client—sometimes referred to as *spoofing the server.* Spoofing makes a server view a client as always connected, even when it is not, thus reducing the number of available licenses. Users can set the duration of IPX watchdog spoofing and periodically disable it so that Novelle NetWare servers can clean up inactive connections.

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. A call made every 5 minutes is not an issue for the other WAN protocols, because they establish dedicated connections rather than establishing connections only as needed.

Use the **ipx watchdog-spoof** command to enable and set the duration of watchdog spoofing. You can specify the number of consecutive hours spoofing is to stay enabled and the number of minutes spoofing is to stay disabled. The server can clean up inactive connections when spoofing is disabled. Be sure that fast switching and autonomous switching are disabled on the serial interface before using this command.

To enable watchdog spoofing, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx watchdog-spoof [enable-time-hours disable-time-minutes]`	Enables and sets the duration of watchdog spoofing.

To keep the serial interface idle when only watchdog packets are being sent, refer to the tasks described in the "Deciding and Preparing to Configure DDR" chapter of the *Cisco IOS Dial Technologies Configuration Guide*. For an example of configuring IPX over DDR, see the "IPX over DDR Example"section in the Novell IPX Configuration Examples chapter.

Configuring 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. Disabling the generation of packets 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 "Deciding and Preparing to Configure DDR" chapter of the *Cisco IOS Dial Technologies Configuration Guide*.

For an example of how to configure SPX spoofing over DDR, see the "IPX over DDR Example" section in the Novell IPX Configuration Examples chapter.

Configuring 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 Medial-Independent PPP and Multilink PPP" chapter in the *Cisco IOS Dial Technologies Configuration Guide*.

Configuring 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 startup 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 on a serial interface, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# no ipx network`	Ensures that you have not configured an IPX network number on the interface.
Step 2	`Router(config-if)# encapsulation ppp`	Enables PPP.
Step 3	`Router(config-if)# ipx ipxwan` [*local-node* {*network-number* | **unnumbered**} *local-server-name retry-interval retry-limit*]	Enables IPXWAN.
Step 4	`Router(config-if)# ipx ipxwan error` [**reset** | **resume** | **shutdown**]	Optionally, defines how to handle IPXWAN when a serial link fails.
Step 5	`Router(config-if)# ipx ipxwan static`	Optionally, enables static routing with IPXWAN. Note that the remote site must also use static routing.

Controlling Access to IPX Networks

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

Types of Access Lists

You can create the following IPX access lists 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 use numbers from 800 to 899 or names to identify them.
- 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 use numbers from 900 to 999 or names to identify them.
- SAP access list—Restricts traffic based on the IPX SAP type. These lists are used for SAP filters and GNS response filters. Novell SAP access lists use numbers from 1000 to 1099 or names to identify them.

- IPX NetBIOS access list—Restricts IPX NetBIOS traffic based on NetBIOS names, not numbers.
- NLSP route aggregation access list—Specifies in detail which routes to summarize and which routes to redistribute explicitly. For more information about route aggregation, see the "Configuring Route Aggregation" section in the Novell IPX Configuration Examples chapter.

Types of Filters

There are more than 14 different IPX filters that you can define for IPX interfaces. They fall into the following six groups:

- Generic filters—Control which data packets are routed in or out of an interface based on the source and destination addresses and IPX protocol type of the packet.
- Routing table filters—Control which 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 GNS response messages it sends out.
- IPX NetBIOS filters—Control incoming and outgoing IPX NetBIOS packets.
- Broadcast filters—Control which broadcast packets are forwarded.
- NLSP route aggregation filters—Control the redistribution of routes and services into and out of an NLSP area.

Table 9 summarizes the filters, the access lists they use, and the commands used to define the filters in the first five groups. Use the **show ipx interfaces** command to display the filters defined on an interface. For additional information about route aggregation, see the "Configuring Route Aggregation" section in the Novell IPX Configuration Examples chapter.

Table 9 IPX Filters

Filter Type	Access List Used by Filter	Command to Define Filter
Generic filters		
Filters inbound or outbound packets based on the contents of the IPX network header.	Standard or Extended	**ipx access-group** {*access-list-number* | *name*} [**in** | **out**]
Routing table filters		
Controls which networks are added to the routing table.	Standard or Extended	**ipx input-network-filter** {*access-list-number* | *name*}
Controls which networks are advertised in routing updates.	Standard or Extended	**ipx output-network-filter** {*access-list-number* | *name*}
Controls which networks are advertised in the Enhanced IGRP routing updates sent out by the Cisco IOS software.	Standard or Extended	**distribute-list** {*access-list-number* | *name*} **out** [*interface-name* | *routing-process*]
Controls the routers from which updates are accepted.	Standard or Extended	**ipx router-filter** {*access-list-number* | *name*}
SAP filters		
Filters incoming service advertisements.	SAP	**ipx input-sap-filter** {*access-list-number* | *name*}
Filters outgoing service advertisements.	SAP	**ipx output-sap-filter** {*access-list-number* | *name*}

Table 9 IPX Filters (continued)

Filter Type	Access List Used by Filter	Command to Define Filter
Controls the routers from which SAP updates are accepted.	SAP	**ipx router-sap-filter** {*access-list-number* | *name*}
Filters list of servers in GNS response messages.	SAP	**ipx output-gns-filter** {*access-list-number* | *name*}
IPX NetBIOS filters		
Filters incoming packets by node name.	IPX NetBIOS	**ipx netbios input-access-filter host** *name*
Filters incoming packets by byte pattern.	IPX NetBIOS	**ipx netbios input-access-filter bytes** *name*
Filters outgoing packets by node name.	IPX NetBIOS	**ipx netbios output-access-filter host** *name*
Filters outgoing packets by byte pattern.	IPX NetBIOS	**ipx netbios output-access-filter bytes** *name*
Broadcast filters		
Controls which broadcast packets are forwarded.	Standard or Extended	**ipx helper-list** {*access-list-number* | *name*}

Implementation Considerations

Remember the following information 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, we recommend 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.
- For numbered access lists, 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. Consequently, 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 reenter it with the new entries.

  For named access lists, all new entries to an existing list are placed at the end of the list. You cannot add entries to the middle of a list. However, you can remove specific entries using the **no deny** and **no permit** commands, rather than deleting the entire access list.
- Do not set up conditions that result in packets getting lost. One way you can lose packets 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 SAP packets within an NLSP area. You can filter them at the boundary of NLSP and RIP/SAP areas, though restrictions do apply. For more information about filtering at these boundaries, see the "Relationship Between Filtering and Route Summarization" section in the Novell IPX Configuration Examples chapter, and the *Novell NetWare Link Services Protocol (NLSP) Specification* publication.

Controlling Access to IPX Networks Task List

To control access to IPX networks, perform the required tasks in the following sections:

- Creating Access Lists (Required)

- Creating Filters (Required)

Creating Access Lists

You can create access lists using numbers or names. You can choose which method you prefer. If you use numbers to identify your access lists, you are limited to 100 access lists per filter type. If you use names to identify your access lists, you can have an unlimited number of access lists per filter type.

The following sections describe how to perform these tasks:

- Creating Access Lists Using Numbers (Optional)
- Creating Access Lists Using Names (Optional)

Creating Access Lists Using Numbers

To create access lists using numbers, use one or more of the following commands in global configuration mode:

Command	Purpose
`Router(config)# access-list access-list-number {deny | permit} source-network [.source-node [source-node-mask]][destination-network [.destination-node [destination-node-mask]]]`	Defines a standard IPX access list using a number. (Generic, routing, and broadcast filters use this type of access list.)
`Router(config)# access-list access-list-number {deny | permit} protocol [source-network [.source-node [source-network-mask.source-node-mask]] source-socket [destination-network [.destination-node [destination-network-mask.destination-node-mask] destination-socket] [log] [time-range time-range-name]`	Defines an extended IPX access list using a number. (Generic, routing, and broadcast filters use this type of access list.) Use the **log** keyword to get access list logging messages, including violations. Specifies a time range to restrict when the **permit** or **deny** statement is in effect.
`Router(config)# access-list access-list-number {deny | permit} network [.node] [network-mask.node-mask] [service-type [server-name]]`	Defines a SAP filtering access list using a number. (SAP and GNS response filters use this type of access list.)

Once you have created an access list using numbers, apply it to the appropriate interfaces using filters as described in the "Creating Filters" section later in this chapter. Applying a filter will activate the access list.

Creating Access Lists Using Names

IPX named access lists allow you to identify IPX access lists with an alphanumeric string (a name) rather than a number. Using IPX named access lists allows you to maintain security by using a separate and easily identifiable access list for each user or interface. IPX named access lists also remove the limit of 100 lists per filter type.You can configure an unlimited number of the following types of IPX named access lists:

- Standard
- Extended

- SAP
- NLSP route aggregation (summarization)
- NetBIOS

If you identify your access list with a name rather than a number, the mode and command syntax are slightly different.

Implementation Considerations

Consider the following information before configuring IPX named access lists:

- Except for NetBIOS access lists, access lists specified by name are not compatible with releases prior to Cisco IOS Release 11.2(4)F.
- Access list names must be unique across all protocols.
- Except for NetBIOS access lists, numbered access lists are also available.

IPX Named Access List Configuration Task List

To configure IPX named access lists for standard, extended, SAP, NLSP route aggregation (summarization), or NetBIOS access lists, perform one or more of the tasks in the following sections:

- Creating a Named Standard Access List (Optional)
- Creating a Named Extended Access List (Optional)
- Creating a Named SAP Filtering Access List (Optional)
- Creating a Named NLSP Route Aggregation Access List (Optional)
- Creating a NetBIOS Access List (Optional)
- Applying Time Ranges to Access Lists (Optional)

Creating a Named Standard Access List

To create a named standard access list, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx access-list standard` *name*	Defines a standard IPX access list using a name. (Generic, routing, and broadcast filters use this type of access list.)
Step 2	`Router(config-access-list)# {deny | permit}` *source-network* `[.`*source-node* `[`*source-node-mask*`]] [`*destination-network* `[.`*destination-node* `[`*destination-node-mask*`]]]`	In access-list configuration mode, specifies one or more conditions allowed or denied. This determines whether the packet is passed or dropped.
Step 3	`Router(config)# exit`	Exits access-list configuration mode.

For an example of creating a named standard access list, see the "Standard Named Access List Example" section in the Novell IPX Configuration Examples chapter.

Creating a Named Extended Access List

To create a named extended access list, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx access-list extended name`	Defines an extended IPX access list using a name. (Generic, routing, and broadcast filters use this type of access list.)
Step 2	`Router(config-access-list)# {deny | permit} protocol [source-network] [[[.source-node] source-node-mask] | [.source-node source-network-mask.source-node-mask]] [source-socket] [destination-network] [[[.destination-node] destination-node-mask] | [.destination-node destination-network-mask.destination-nodemask]] [destination-socket] [log] [time-range time-range-name]`	In access-list configuration mode, specifies the conditions allowed or denied. Use the **log** keyword to get access list logging messages, including violations. Specifies a time range to restrict when the **permit** or **deny** statement is in effect.
Step 3	`Router(config)# exit`	Exits access-list configuration mode.

Creating a Named SAP Filtering Access List

To create a named access list for filtering SAP requests, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx access-list sap name`	Defines a SAP filtering access list using a name. (SAP, GNS, and Get General Service (GGS) response filters use this type of access list.)
Step 2	`Router(config-access-list)# {deny | permit} network [.node] [network-mask.node-mask] [service-type [server-name]]`	In access-list configuration mode, specifies the conditions allowed or denied.
Step 3	`Router(config)# exit`	Exits access-list configuration mode.

Creating a Named NLSP Route Aggregation Access List

NLSP route aggregation access lists perform one of the following functions:

- Permit networks to be redistributed as explicit networks, without summarization.
- Deny the redistribution of explicit networks and generate an appropriate aggregating (summary) route for redistribution.

To create a named access list for NLSP route aggregation, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx access-list summary name`	Defines an IPX access list for NLSP route aggregation using a name.
Step 2	`Router(config-access-list)# {deny | permit} network network-mask [ticks ticks] [area-count area-count]`	In access-list configuration mode, specifies the conditions allowed or denied. For each address range you want to redistribute as a single aggregated route, use the **deny** keyword. For each address that you want to redistribute explicitly, use the **permit** keyword.
Step 3	`Router(config)# exit`	Exits access-list configuration mode.

For information on how to use named access list when configuring route aggregation, see the tasks listed in the "Route Aggregation Configuration Task List" section in the Novell IPX Configuration Examples chapter.

Creating a NetBIOS Access List

To create a NetBIOS access list, use one or more of the following commands in global configuration mode:

Command	Purpose
`Router(config)# netbios access-list host name {deny | permit} string`	Creates an access list for filtering IPX NetBIOS packets by node name. (NetBIOS filters use this type of access list.)
`Router(config)# netbios access-list bytes name {deny | permit} offset byte-pattern`	Creates an access list for filtering IPX NetBIOS packets by arbitrary byte pattern. (NetBIOS filters use this type of access list.)

Modifying IPX Named Access Lists

After you initially create an access list, you place any subsequent additions (possibly entered from the terminal) at the end of the list. In other words, you cannot selectively add access list command lines to the middle of a specific access list. However, you can use **no permit** and **no deny** commands to remove entries from a named access list.

Note When creating access lists, remember that, by default, the end of the access list contains an implicit deny statement for everything if it did not find a match before reaching the end.

For an example of creating a generic filter, see the "IPX Network Access Examples" section in the Novell IPX Configuration Examples chapter.

Applying Named Access Lists to Interfaces

After creating an access list, you must apply it to the appropriate interface using filters as described in the "Creating Filters" section later in this chapter. Applying a filter will activate the access list.

Applying Time Ranges to Access Lists

It is now possible to implement access lists based on the time of day and week using the **time-range** command. To do so, first define the name of the time range and times of the day and week, then reference the time range by name in an access list to apply the restrictions of the time range to the access list.

Currently, IP and IPX named or numbered extended access lists are the only functions that can use time ranges. The time range allows the network administrator to define when the **permit** or **deny** statements in the access list are in effect. Prior to this time range feature, access list statements were always in effect once they were applied. The **time-range** keyword and argument are referenced in the named and numbered extended access list task tables in the previous sections, "Creating Access Lists Using Numbers" and "Creating Access Lists Using Names." The **time-range** command is configured in the "Performing Basic System Management" chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide*. See the "IPX Network Access Examples" section in the Novell IPX Configuration Examples chapter, for a configuration example of IPX time ranges.

There are many possible benefits of time ranges, such as the following:

- The network administrator has more control over permitting or denying a user access to resources. These resources could be an application (identified by an IP address/mask pair and a port number), policy routing, or an on-demand link (identified as interesting traffic to the dialer).
- Network administrators can set time-based security policy, including:
  - Perimeter security using the Cisco IOS Firewall feature set or access lists
  - Data confidentiality with Cisco Encryption Technology or IPS
- Policy-based routing and queueing functions are enhanced.
- When provider access rates vary by time of day, it is possible to automatically reroute traffic cost effectively.
- Service providers can dynamically change a committed access rate (CAR) configuration to support the quality of service (QoS) Service Level Agreements (SLAs) that are negotiated for certain times of day.
- Network administrators can control logging messages. Access list entries can log traffic at certain times of the day, but not constantly. Therefore, administrators can simply deny access without needing to analyze many logs generated during peak hours.

Creating Filters

Filters allow you to control which traffic is forwarded or blocked at the interfaces of the router. Filters apply specific numbered or named access lists to interfaces.

To create filters, perform the tasks in the following sections:

- Creating Generic Filters (Optional)
- Creating Filters for Updating the Routing Table (Optional)
- Creating SAP Filters (Optional)
- Creating GNS Response Filters (Optional)
- Creating GGS Response Filters (Optional)
- Creating IPX NetBIOS Filters (Optional)
- Creating Broadcast Message Filters (Optional)

Creating Generic Filters

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

To create generic filters, first create a standard or an extended access list as described in the "Creating Access Lists" section earlier in this chapter and then apply a filter to an interface.

To apply a generic filter to an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx access-group {access-list-number | name}[in | out]`	Applies a generic filter to an interface.

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 Examples" section in the Novell IPX Configuration Examples chapter.

Creating 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, first create a standard or an extended access list as described in the "Creating Access Lists" section earlier in this chapter and then apply one or more routing filters to an interface.

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

Command	Purpose
`Router(config-if)# ipx input-network-filter {access-list-number | name}`	Controls which networks are added to the routing table when IPX routing updates are received.
`Router(config-if)# ipx output-network-filter {access-list-number | name}`	Controls which networks are advertised in RIP routing updates sent out by the Cisco IOS software.
`Router(config-router)# distribute-list {access-list-number | name} out [interface-name | routing-process]`	Controls which networks are advertised in the Enhanced IGRP routing updates sent out by the Cisco IOS software.
`Router(config-if)# ipx router-filter {access-list-number | name}`	Controls the routers from which routing updates are accepted.

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 "Controlling the Advertising of Routes in Routing Updates" section earlier in this chapter for more information.

Creating 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, first create a SAP filtering access list as described in the "Creating Access Lists" section earlier in this chapter and then apply one or more filters to an interface.

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

Command	Purpose
`Router(config-if)# ipx input-sap-filter {access-list-number | name}`	Filters incoming service advertisements.
`Router(config-if)# ipx output-sap-filter {access-list-number | name}`	Filters outgoing service advertisements.
`Router(config-if)# ipx router-sap-filter {access-list-number | name}`	Filters service advertisements received from a particular router.

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 in the Novell IPX Configuration Examples chapter.

Creating GNS Response Filters

To create filters for controlling which servers are included in the GNS responses sent by the Cisco IOS software, first create a SAP filtering access list as described in the "Creating Access Lists" section earlier in this chapter and then apply a GNS filter to an interface.

To apply a GNS filter to an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx output-gns-filter {access-list-number | name}`	Filters the list of servers in GNS response messages.

Creating GGS Response Filters

To create filters for controlling which servers are included in the Get General Service (GGS) responses sent by the Cisco IOS software, first create a SAP filtering access list as described in the "Creating Access Lists" section earlier in this chapter and then apply a GGS filter to an interface.

Note Because GGS SAP response filters are applied ahead of output SAP filters, a SAP entry permitted to pass through the GGS SAP response filter can still be filtered by the output SAP filter.

To apply a GGS filter to an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx output-ggs-filter`	Filters the list of servers in GGS response messages.

For an example of creating a GGS SAP response filter, see the "IPX Network Access Examples"section in the Novell IPX Configuration Examples chapter.

Creating IPX NetBIOS Filters

The Novell 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 Logic Link Control, type 2 (LLC2) NetBIOS packets.

Implementation Considerations

Remember the following 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 nodes are filtered by 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.

Configuring IPX NetBIOS Filters

To create filters for controlling IPX NetBIOS access, first create a NetBIOS access list as described in the "Creating Access Lists" section in the Novell IPX Configuration Examples chapter, and then apply the access list to an interface.

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

Command	Purpose
`Router(config-if)# ipx netbios input-access-filter host name`	Filters incoming packets by node name.
`Router(config-if)# ipx netbios input-access-filter bytes name`	Filters incoming packets by byte pattern.
`Router(config-if)# ipx netbios output-access-filter host name`	Filters outgoing packets by node name.
`Router(config-if)# ipx netbios output-access-filter bytes name`	Filters outgoing packets by byte pattern.

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 in the Novell IPX Configuration Examples chapter.

Creating Broadcast Message Filters

Routers normally block all broadcast requests and do not forward them to other network segments, therefore preventing 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, first create a standard or an extended access list as described in the "Creating Access Lists" section earlier in this chapter and then apply a broadcast message filter to an interface.

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

	Command	Purpose
Step 1	`Router(config-if)# ipx helper-address network.node`	Specifies a helper address for forwarding broadcast messages.
Step 2	`Router(config-if)# ipx helper-list {access-list-number | name}`	Applies a broadcast message filter to an interface.

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 in the Novell IPX Configuration Examples chapter.

Tuning IPX Network Performance

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

- Controlling Novell IPX Compliance (Optional)
- Adjusting RIP and SAP Information (Optional)
- Configuring Load Sharing (Optional)
- Specifying the Use of Broadcast Messages (Optional)
- Disabling IPX Fast Switching (Optional)
- Adjusting the Route Cache (Optional)
- Adjusting Default Routes (Optional)
- Padding Odd-Length Packets (Optional)

Controlling Novell IPX Compliance

The Cisco implementation of the Novell IPX protocol is certified to provide full IPX router functionality, as defined by the Novell *IPX Router Specification, version 1.10* publication published November 17, 1992.

To control compliance to Novell specifications, perform the tasks in the following sections:

- Controlling the Forwarding of Type 20 Packets (Optional)
- Controlling Interpacket Delay (Optional)
- Shutting Down an IPX Network (Optional)
- Achieving Full Novell Compliance (Optional)

Controlling 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 packets.

How Type 20 Packet Propagation Works

When an interface configured for Type 20 propagation receives a Type 20 packet, Cisco IOS software processes the packet according to Novell specifications. Cisco IOS software propagates the packet to the next interface. The Type 20 packet can be propagated for up to eight hop counts.

Loop Detection and Other Checks

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.

Although this extra checking increases the robustness of Type 20 propagation packet handling by decreasing the amount of unnecessary packet replication, it has the following 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.

Relationship Between Type 20 Propagation and Helper Addresses

You use helper addresses to forward non-Type 20 broadcast packets to other network segments. For information on forwarding other broadcast packets, see the "Using Helper Addresses to Forward Broadcast Packets" section later in this chapter.

You can use helper addresses and Type 20 propagation together in your network. Use helper addresses to forward non-Type 20 broadcast packets and use Type 20 propagation to forward Type 20 broadcast packets.

Type 20 Packets Configuration Task List

You can enable the forwarding of Type 20 packets on individual interfaces. Additionally, you can restrict the acceptance and forwarding of Type 20 packets. You can also choose to not comply with Novell specifications and forward Type 20 packets using helper addresses rather than using Type 20 propagation. The following sections describe these tasks:

- Enabling the Forwarding of Type 20 Packets (Optional)
- Restricting the Acceptance of Incoming Type 20 Packets (Optional)
- Restricting the Forwarding of Outgoing Type 20 Packets (Optional)
- Forwarding Type 20 Packets Using Helper Addresses (Optional)

Enabling 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, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx type-20-propagation`	Forwards IPX Type 20 propagation packet broadcasts to other network segments.

When you enable Type 20 propagation, Cisco IOS propagates the broadcast to the next interface up to eight hops.

Restricting 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, which 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, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx type-20-input-checks`	Restricts the acceptance of IPX Type 20 propagation packets.

Restricting 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, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx type-20-output-checks`	Restricts the forwarding of IPX Type 20 propagation packets.

Forwarding 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.

You may want to forward Type 20 packets using helper addresses when some routers in your network are running versions of Cisco IOS that do not support Type 20 propagation. When some routers in your network support Type 20 propagation and others do not, you can avoid flooding packets everywhere in the network by using helper addresses to direct packets to certain segments only.

Cisco IOS Release 9.1 and earlier versions do not support Type 20 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, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx type-20-helpered`	Forwards IPX Type 20 packets to specific networks segments. This step turns off Type 20 propagation.
Step 2	`Router(config-if)# ipx helper-address` *network.node*	From interface configuration mode, specifies a helper address for forwarding broadcast messages, including IPX Type 20 packets.

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.

Controlling Interpacket Delay

To control interpacket delay, you can use a combination of global configuration and interface configuration commands.

Use one or more of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ipx default-output-rip-delay` *delay*	Sets the interpacket delay of multiple-packet routing updates sent on all interfaces.
`Router(config)# ipx default-triggered-rip-delay` *delay*	Sets the interpacket delay of multiple-packet triggered routing updates sent on all interfaces.
`Router(config)# ipx default-output-sap-delay` *delay*	Sets the interpacket delay of multiple-packet SAP updates sent on all interfaces.
`Router(config)# ipx default-triggered-sap-delay` *delay*	Sets the interpacket delay of multiple-packet triggered SAP updates sent on all interfaces.

Use one or more of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx output-rip-delay` *delay*	Sets the interpacket delay of multiple-packet routing updates sent on a single interface.
`Router(config-if)# ipx triggered-rip-delay` *delay*	Sets the interpacket delay of multiple-packet triggered routing updates sent on a single interface.
`Router(config-if)# ipx output-sap-delay` *delay*	Sets the interpacket delay of multiple-packet SAP updates sent on a single interface.
`Router(config-if)# ipx triggered-sap-delay` *delay*	Sets the interpacket delay of multiple-packet triggered SAP updates sent on a single interface.

Note We recommend that you use the **ipx output-rip-delay** and **ipx output-sap-delay** commands on slower speed WAN interfaces. The default delay for Cisco IOS Release 11.1 and later versions is 55 milliseconds.

Shutting Down an IPX Network

To shut down an IPX network using a Novell-compliant method, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx down network`	Administratively shuts down an IPX network on an interface. This removes the network from the interface.

Convergence is faster when you shut down an IPX network using the **ipx down** command than when using the **shutdown** command.

Achieving Full Novell Compliance

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

	Command	Purpose
Step 1	`Router(config-if)# ipx output-rip-delay 55`	Sets the interpacket delay of multiple-packet routing updates to 55 milliseconds.
Step 2	`Router(config-if)# ipx output-sap-delay 55`	Sets the interpacket delay of multiple-packet SAP updates to 55 milliseconds.
Step 3	`Router(config-if)# ipx type-20-propagation`	Optionally enables Type 20 packet propagation if you want to forward Type 20 broadcast traffic across the router.

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 set these interpacket delays, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx default-output-rip-delay 55`	Sets the interpacket delay of multiple-packet routing updates sent on all interfaces to 55 milliseconds.
Step 2	`Router(config)# ipx default-output-sap-delay 55`	Sets the interpacket delay of multiple-packet SAP updates sent on all interfaces to 55 milliseconds.

Note The default delay for Cisco IOS Release 11.1 and later versions is 55 milliseconds.

Adjusting RIP and SAP Information

To adjust RIP and SAP information, perform one or more of the optional tasks in the following sections:

- Configuring Static Routes (Optional)
- Adjusting the RIP Delay Field (Optional)
- Controlling Responses to RIP Requests (Optional)
- Adjusting RIP Update Timers (Optional)
- Configuring RIP Update Packet Size (Optional)
- Configuring Static SAP Table Entries (Optional)
- Configuring the Queue Length for SAP Requests (Optional)
- Adjusting SAP Update Timers (Optional)
- Configuring SAP Update Packet Size (Optional)
- Enabling SAP-after-RIP (Optional)
- Disabling Sending of General RIP or SAP Queries (Optional)
- Controlling Responses to GNS Requests (Optional)

Configuring 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, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx route {network [network-mask] | default} {network.node | interface}[ticks] [hops]`	Adds a static route to the routing table.

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, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx route {network [network-mask] | default} {network.node | interface}[ticks] [hops] [floating-static]`	Adds a floating static route to the routing table.

Adjusting the RIP Delay Field

By default, all LAN interfaces have a RIP delay of 1 and all WAN interfaces have a RIP delay of 6. Leaving the delay at its default value is sufficient for most interfaces. However, you can adjust the RIP delay field by setting the tick count. To set the tick count, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx delay ticks`	Sets the tick count, which is used in the IPX RIP delay field.

Controlling Responses to RIP Requests

To control responses to RIP requests, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx rip-response-delay ms`	Sets the delay when responding to RIP requests.

Adjusting 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 timers only in a configuration in which all routers are Cisco routers, or in which the IPX routers allow configurable timers. The timers should be the same for all devices connected to the same cable segment. The 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 a timer for more than one interface in a router, the granularity of the 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, refer to the "Novell IPX Commands" chapter in the *Cisco IOS AppleTalk and Novell IPX Command Reference.*

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 timers on a per-interface basis, use one or all of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx update interval {rip | sap} {value | changes-only}`	Adjusts the RIP update timer.
`Router(config-if)# ipx output-rip-delay delay`	Adjusts the delay between multiple-packet routing updates sent on a single interface.
`Router(config-if)# ipx triggered-rip-delay delay`	Adjusts the delay between multiple-packet triggered routing updates sent on a single interface.

To adjust RIP update timers on a global basis, use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ipx default-output-rip-delay delay`	Adjusts the delay between multiple-packet routing updates sent on all interfaces.
`Router(config)# ipx default-triggered-rip-delay delay`	Adjusts the delay between multiple-packet triggered routing updates sent on all interfaces.

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

Command	Purpose
`Router(config-if)# ipx rip-multiplier multiplier`	Configures the interval at which a network RIP entry ages out.

Configuring 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, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx rip-max-packetsize bytes`	Configures the maximum packet size of RIP updates sent out an interface.

Configuring 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, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx sap service-type name network.node socket hop-count`	Specifies a static SAP table entry.

Configuring the Queue Length for SAP Requests

The Cisco IOS software maintains a list of SAP requests to process, including all pending Get Nearest Server (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, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx sap-queue-maximum number`	Configures the maximum SAP queue length.

Adjusting 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 such as slower-speed interfaces. You should ensure that all IPX 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.

It is not possible to change the interval at which SAP updates are sent on most PC-based servers. Therefore, you should never change the interval for an Ethernet or Token Ring network that has servers on it.

You can set the router to send an update only when changes have occurred. Using the **changes-only** keyword specifies the sending of a SAP update only when the link comes up, when the link is downed administratively, or when the databases change. The **changes-only** keyword causes the router to do the following:

- Send a single, full broadcast update when the link comes up
- Send appropriate triggered updates when the link is shut down
- Send appropriate triggered updates when specific service information changes

To modify the SAP update timers on a per-interface basis, use one or all of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx update interval {rip | sap} {value | changes-only}`	Adjusts the interval at which SAP updates are sent.
`Router(config-if)# ipx output-sap-delay delay`	Adjusts the interpacket delay of multiple-packet SAP updates sent on a single interface.
`Router(config-if)# ipx triggered-sap-delay delay`	Adjusts the interpacket delay of multiple-packet triggered SAP updates sent on a single interface.

To adjust SAP update timers on a global basis (eliminating the need to configure delays on a per-interface basis), use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ipx default-output-sap-delay delay`	Adjusts the interpacket delay of multiple-packet SAP updates sent on all interfaces.
`Router(config)# ipx default-triggered-sap-delay delay`	Adjusts the interpacket delay of multiple-packet triggered SAP updates sent on all interfaces.

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, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx sap-multiplier multiplier`	Configures the interval at which the SAP entry of a network or server ages out.

Configuring SAP Update Packet Size

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

Command	Purpose
`Router(config-if)# ipx sap-max-packetsize bytes`	Configures the maximum packet size of SAP updates sent out an interface.

Enabling SAP-after-RIP

The IPX SAP-after-RIP feature links SAP updates to RIP updates so that SAP broadcast and unicast updates automatically occur immediately after the completion of the corresponding RIP update. This feature ensures that a remote router does not reject service information because it lacks a valid route to the service. As a result of this feature, periodic SAP updates are sent at the same interval as RIP updates.

The default behavior of the router is to send RIP and SAP periodic updates with each using its own update interval, depending on the configuration. In addition, RIP and SAP periodic updates are jittered slightly, such that they tend to diverge from each other over time. This feature synchronizes SAP and RIP updates.

Sending all SAP and RIP information in a single update reduces bandwidth demands and eliminates erroneous rejections of SAP broadcasts.

Linking SAP and RIP updates populates the service table of the remote router more quickly, because services will not be rejected due to the lack of a route to the service. Populating the service table more quickly can be especially useful on WAN circuits where the update intervals have been greatly increased to reduce the overall level of periodic update traffic on the link.

To configure the router to send a SAP update following a RIP broadcast, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx update sap-after-rip`	Configures the router to send a SAP broadcast immediately following a RIP broadcast.

Disabling Sending of General RIP or SAP Queries

You can disable the sending of general RIP or SAP queries on a link when it first comes up to reduce traffic and save bandwidth.

RIP and SAP general queries are normally sent by remote routers when a circuit first comes up. On WAN circuits, two full updates of each kind are often sent across the link. The first update is a full broadcast update, triggered locally by the link-up event. The second update is a specific (unicast) reply triggered by the general query received from the remote router. If you disable the sending of general queries when the link first comes up, it is possible to reduce traffic to a single update, and save bandwidth.

To disable the sending of a general RIP or SAP query when an interface comes up, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no ipx linkup-request {rip | sap}`	Disables the sending of a general RIP or SAP Query when an interface comes up.

To reenable the sending of a general RIP or SAP query, use the positive form of the command.

Controlling 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, use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ipx gns-round-robin`	Responds to GNS requests using a round-robin selection method.
`Router(config)# ipx gns-response-delay [milliseconds]`	Sets the delay when responding to GNS requests.

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.

To disable GNS queries on a per-interface basis, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx gns-reply-disable`	Disables the sending of replies to Get Nearest Server (GNS) queries.

Configuring Load Sharing

To configure IPX to perform round-robin or per-host load sharing, perform the tasks described in the following sections:

- Enabling Round-Robin Load Sharing (Optional)
- Enabling per-Host Load Sharing (Optional)

Enabling 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, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx maximum-paths paths`	Sets the maximum number of equal-cost paths to a destination.

Enabling 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 increases transmission speed, 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, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx maximum-paths paths`	Sets the maximum number of equal cost paths to a destination to a value greater than 1.
Step 2	`Router(config)# ipx per-host-load-share`	Enables per-host load sharing.

Specifying the Use of Broadcast Messages

To specify the use of broadcast messages, perform the tasks described in the following sections:

- Using Helper Addresses to Forward Broadcast Packets (Optional)
- Enabling Fast Switching of IPX Directed Broadcast Packets (Optional)

Using Helper Addresses to Forward Broadcast Packets

Routers normally block all broadcast requests and do not forward them to other network segments, therefore preventing the degradation of performance over the entire network. However, you can enable the router to forward broadcast packets to helper addresses on other network segments.

How Helper Addresses Work

Helper addresses specify the network and node on another segment that can receive unrecognized broadcast packets. Unrecognized broadcast packets are non-RIP and non-SAP packets that are not addressed to the local network.

When the interface configured with helper addresses receives an unrecognized broadcast packet, Cisco IOS software changes the broadcast packet to a unicast and sends the packet to the specified network and node on the other network segment. Unrecognized broadcast packets are not flooded everywhere in your network.

With helper addresses, there is no limit on the number of hops that the broadcast packet can make.

Fast Switching Support

Cisco IOS supports fast switching of helpered broadcast packets.

When to Use Helper Addresses

You use helper addresses when you want to forward broadcast packets (except Type 20 packets) to other network segments.

Forwarding broadcast packets to helper addresses 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 packet.

Relationship Between Helper Addresses and Type 20 Propagation

You use Type 20 packet propagation to forward Type 20 packets to other network segments. For information on forwarding Type 20 packets, see the "Controlling the Forwarding of Type 20 Packets" section earlier in this chapter.

You can use helper addresses and Type 20 propagation together in your network. Use helper addresses to forward non-Type 20 broadcast packets and use Type 20 propagation to forward Type 20 broadcast packets.

Implementation Considerations

Using helper addresses is not Novell-compliant. However, it does allow routers to forward broadcast packets to network segments that can process them without flooding the network. It also allows routers running versions of Cisco IOS that do not support Type 20 propagation to forward Type 20 packets.

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.

Using Helper Addresses

To specify a helper address for forwarding broadcast packets, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx helper-address` *network.node*	Specifies a helper address for forwarding broadcast messages.

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 in the Novell IPX Configuration Examples chapter.

Enabling 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, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx broadcast-fastswitching`	Enables fast switching of IPX-directed broadcast packets.

Disabling IPX Fast Switching

By default, fast switching is enabled on all interfaces that support 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, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no ipx route-cache`	Disables IPX fast switching.

Adjusting the Route Cache

Adjusting the route cache allows you to control the size of the route cache, reduce memory consumption, and improve router performance. You accomplish these tasks by controlling the route cache size and invalidation. The following sections describe these optional tasks:

- Controlling Route Cache Size (Optional)
- Controlling Route Cache Invalidation (Optional)

Controlling 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, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx route-cache max-size` *size*	Sets a maximum limit on the number of entries in the IPX route cache.

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 "Controlling Route Cache Invalidation" section later in this chapter for more information on invalidating route cache entries.

Controlling Route Cache Invalidation

You can configure the router to invalidate fast-switch cache entries that are inactive. If these entries remain invalidated for 1 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, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx route-cache inactivity-timeout` *period* [*rate*]	Invalidates fast-switch cache entries that are inactive.

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.

Adjusting Default Routes

You can adjust the use of default routes in your IPX network. You can turn off the use of network number -2 as the default route. You can also specify that the router advertise only default RIP routes out an interface. The following sections describe these optional tasks:

- Disabling Network Number -2 as the Default Route (Optional)
- Advertising Only Default RIP Routes (Optional)

Disabling Network Number -2 as the Default Route

The default route is used when a route to any destination network is unknown. All packets for which a route to the destination address is unknown are forwarded to the default route. By default, IPX treats network number -2 (0xFFFFFFFE) as the default route.

For an introduction to default routes, see the "IPX Default Routes" section earlier in this chapter. For more background information on how to handle IPX default routes, refer to the Novell *NetWare Link Services Protocol (NLSP) Specification, Revision 1.1* publication.

By default, Cisco IOS software treats network -2 as the default route. You can disable this default behavior and use network -2 as a regular network number in your network.

To disable the use of network number -2 as the default route, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no ipx default-route`	Disables default route handling.

Advertising Only Default RIP Routes

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, therefore greatly reducing the CPU overhead when routing tables are large.

To advertise only the default route via an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx advertise-default-route-only network`	Advertises only the default RIP route.

Padding Odd-Length Packets

Some IPX end hosts accept only even-length Ethernet packets. If the length of a packet is odd, the packet must be padded with an extra byte so that end host can receive it. By default, Cisco IOS pads odd-length Ethernet packets.

However, there are cases in certain topologies where nonpadded Ethernet packets are forwarded onto a remote Ethernet network. Under specific conditions, you can enable padding on intermediate media as a temporary workaround for this problem. Note that you should perform this task only under the guidance of a customer engineer or other service representative.

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

	Command	Purpose
Step 1	`Router(config-if)# no ipx route-cache`	Disables fast switching.
Step 2	`Router(config-if)# ipx pad-process-switched-packets`	Enables the padding of odd-length packets.

Shutting 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, therefore allowing the neighboring systems to update their routing, SAP, and other tables without needing 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, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx down network`	Shuts down an IPX network, but allows the network to still exist in the configuration.

To shut down an IPX network and remove it from the configuration, use one of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# no ipx network`	Shuts down an IPX network and removes it from the configuration.
`Router(config-if)# no ipx network` *network* (where *network* is 1, the primary interface)	When multiple networks are configured on an interface, shuts down all networks and removes them from the interface.
`Router(config-if)# no ipx network` *network* (where *network* is the number of the secondary interface [not 1])	When multiple networks are configured on an interface, shuts down one of the secondary networks and removes it from the interface.

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

For an example of shutting down an IPX network, see the "IPX Routing Examples"section in the Novell IPX Configuration Examples chapter.

Configuring IPX Accounting

IPX accounting enables 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. IPX accounting tracks only IPX traffic that is routed out an interface on which IPX accounting is configured; it does not track traffic generated by or terminated at the router itself.

The Cisco IOS software maintains two accounting databases: an active database and a checkpoint database. The active database contains accounting data tracked until the database is cleared. When the active database is cleared, its contents are copied to the checkpoint database. Using these two databases together enables you to monitor both current traffic and traffic that has previously traversed the router.

Switching Support

Process and fast switching support IPX accounting statistics. Autonomous and silicon switching engine (SSE) switching do not support IPX accounting statistics.

Note CiscoBus (Cbus) and SSE are not supported on the MIP interface.

Access List Support

IPX access lists support IPX accounting statistics.

IPX Accounting Task List

To configure IPX accounting, perform the tasks in the following sections. The first task is required; the remaining task is optional.

- Enabling IPX Accounting (Required)
- Customizing IPX Accounting (Optional)

Enabling IPX Accounting

To enable IPX accounting, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx accounting`	Enables IPX accounting.

Customizing IPX Accounting

To customize IPX accounting, use one or more of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ipx accounting-threshold` *threshold*	Sets the maximum number of accounting entries.
`Router(config)# ipx accounting-transits` *count*	Sets the maximum number of transit entries.
`Router(config)# ipx accounting-list` *number mask*	Defines the filter networks for which IPX accounting information is kept. Use one command for each network.

Transit entries are entries in the database that do not match any of the networks specified by the **ipx accounting-list** commands.

If you enable IPX accounting on an interface but do not specify an accounting list, IPX accounting tracks all traffic through the interface (all transit entries) up to the accounting threshold limit.

For an example of how to configure IPX accounting, see the “IPX Accounting Example”section in the Novell IPX Configuration Examples chapter.

Configuring IPX Between LANs

Cisco IOS software supports routing IPX between Ethernet-emulated LANs and Token Ring-emulated LANs. For more information on emulated LANs and routing IPX between them, refer to the “Configuring LAN Emulation” chapter of the *Cisco IOS Switching Services Configuration Guide*.

Configuring IPX Between VLANs

Cisco IOS software supports routing IPX between VLANs. Users with Novell NetWare environments can configure any one of the four IPX Ethernet encapsulations to be routed using the Inter-Switch Link (ISL) encapsulation across VLAN boundaries. For more information on VLANs and routing IPX between them over ISL, refer to the "Configuring Routing Between VLANs with ISL Encapsulation" chapter of the *Cisco IOS Switching Services Configuration Guide*.

Configuring IPX Multilayer Switching

Cisco IOS software supports IPX Multilayer Switching (MLS). For more information on IPX MLS, refer to the "Multilayer Switching" chapter of the *Cisco IOS Switching Services Configuration Guide*.

Monitoring and Maintaining the IPX Network

To monitor and maintain your IPX network, perform the optional tasks described in the following sections:

- General Monitoring and Maintaining Tasks (Optional)
- Monitoring and Maintaining IPX Enhanced IGRP (Optional)
- Monitoring and Maintaining NLSP (Optional)
- Monitoring and Maintaining NHRP (Optional)
- Monitoring and Maintaining IPX Accounting (Optional)

General Monitoring and Maintaining Tasks

You can perform one or more of these general monitoring and maintaining tasks as described in the following sections:

- Monitoring and Maintaining Caches, Tables, Interfaces, and Statistics (Optional)
- Specifying the Type and Use of Ping Packets (Optional)
- Troubleshooting Network Connectivity (Optional)

Monitoring and Maintaining Caches, Tables, Interfaces, and Statistics

To monitor and maintain caches, tables, interfaces, or statistics in a Novell IPX network, use one or more of the following commands in EXEC mode:

Command	Purpose
Router> **clear ipx cache**	Deletes all entries in the IPX fast-switching cache.
Router> **clear ipx route** [*network* | *****]	Deletes entries in the IPX routing table.
Router> **clear ipx traffic**	Clears IPX traffic counters.
Router> **show ipx cache**	Lists the entries in the IPX fast-switching cache.
Router> **show ipx interface** [*type number*]	Displays the status of the IPX interfaces configured in the router and the parameters configured on each interface.
Router> **show ipx route** [*network*] [**default**] [**detailed**]	Lists the entries in the IPX routing table.
Router> **show ipx servers** [**unsorted** | **sorted** [**name** | **net** | **type**]] [**regexp** *name*]	Lists the servers discovered through SAP advertisements.
Router> **show ipx traffic** [**since** {**bootup** | **show**}]	Displays information about the number and type of IPX packets sent and received.
Router> **show sse summary**	Displays a summary of SSE statistics.

Specifying the Type and Use of Ping Packets

The Cisco IOS software can send Cisco pings and standard Novell pings as defined in the NLSP specification or diagnostic request packets. By default, the software generates Cisco pings. To choose the ping type, use the following command in global configuration mode:

Command	Purpose
Router(config)# **ipx ping-default** {**cisco** | **novell** | **diagnostic**}	Selects the ping type.

The IPX diagnostic ping feature addresses diagnostic related issues by accepting and processing unicast or broadcast diagnostic packets. It makes enhancements to the current IPX **ping** command to ping other stations using the diagnostic packets and display the configuration information in the response packet.

Note When a ping is sent from one station to another, the response is expected to come back immediately; when the **ipx ping-default** command is set to diagnostics, the response could consist of more than one packet and each node is expected to respond within 0.5 seconds of receipt of the request. Due to the absence of an end-of-message flag, there is a delay and the requester must wait for all responses to arrive. Therefore, in verbose mode there may be a brief delay of 0.5 seconds before the response data is displayed.

The **ipx ping** command using the **diagnostic** keyword can be used to conduct a reachability test and should not be used to measure accurate round-trip delay.

To initiate a ping, use one of the following commands in EXEC mode:

Command	Purpose
`Router# ping ipx network.node`	Diagnoses basic IPX network connectivity (user-level command).
`Router# ping [ipx] [network.node]`	Diagnoses basic IPX network connectivity (privileged command).

Troubleshooting Network Connectivity

To trace the IPX destination and measure roundtrip delays, use the following command in either user or privileged EXEC mode:

Command	Purpose
`Router> trace [protocol] [destination]`	Traces packet routes through the network (user or privileged).

Note In user EXEC mode, you are not allowed to change the trace route timeout interval, probe count, minimum and maximum time to live, and verbose mode. To do so, use the **trace** command in privileged EXEC mode.

Monitoring and Maintaining IPX Enhanced IGRP

To monitor and maintain Enhanced IGRP on an IPX network, use one or more of the following commands in EXEC mode:

Command	Purpose
`Router> show ipx eigrp neighbors [servers] [autonomous-system-number | type number [regexp name]]`	Lists the neighbors discovered by IPX Enhanced IGRP.
`Router> show ipx eigrp interfaces [type number] [as-number]`	Displays information about interfaces configured for Enhanced IGRP.
`Router> show ipx eigrp topology [network]`	Displays the contents of the IPX Enhanced IGRP topology table.
`Router> show ipx route [network]`	Displays the contents of the IPX routing table, including Enhanced IGRP entries.
`Router> show ipx traffic`	Displays information about IPX traffic, including Enhanced IGRP traffic.

Logging 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, use the following command in IPX-router configuration mode:

Command	Purpose
`Router(config-ipx-router)# log-neighbor-changes`	Enables logging of Enhanced IGRP neighbor adjacency changes.

Monitoring and Maintaining NLSP

To monitor and maintain NLSP on an IPX network, use one or more of the following commands in EXEC mode:

Command	Purpose
`Router> clear ipx nlsp [tag] neighbors`	Deletes all NLSP adjacencies from the adjacency database.
`Router> clear ipx nlsp traffic`	Clears NLSP traffic counters.
`Router> show ipx nlsp [tag] database [lspid] [detail]`	Displays the entries in the LSP database.
`Router> show ipx nlsp [tag] neighbors [interface] [detail]`	Displays the NLSP neighbors of the device and their states.
`Router> show ipx nlsp [tag] spf-log`	Displays a history of the SPF calculations for NLSP.
`Router> show ipx nlsp traffic [since {bootup | show}]`	Displays cumulative traffic statistics for NLSP traffic counters.

Logging Adjacency State Changes

You can allow NLSP to generate a log message when an NLSP adjacency changes state (up or down). Generating a log message may be very useful when monitoring large networks. Messages are logged using the system error message facility. Messages are in the following 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, use the following command in IPX-router configuration mode:

Command	Purpose
`Router(config-ipx-router)# log-adjacency-changes`	Logs NLSP adjacency state changes.

Monitoring and Maintaining NHRP

To monitor the NHRP cache or traffic, use either of the following commands in EXEC mode:

Command	Purpose
Router> **show ipx nhrp** [**dynamic** | **static**] [*type number*]	Displays the IPX NHRP cache, optionally limited to dynamic or static cache entries for a specific interface.
Router> **show ipx nhrp traffic**	Displays NHRP traffic statistics.

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, use the following command in EXEC mode:

Command	Purpose
Router> **clear ipx nhrp**	Clears the IPX NHRP cache of dynamic entries.

Monitoring and Maintaining IPX Accounting

To monitor and maintain IPX accounting in your IPX network, use the following commands in EXEC mode:

Command	Purpose
Router> **clear ipx accounting** [**checkpoint**]	Deletes all entries in the IPX accounting or accounting checkpoint database.
Router> **show ipx accounting** [**checkpoint**]	Lists the entries in the IPX accounting or accounting checkpoint database.