CHAPTER 20

Configuring Novell IPX

Novell Internet Packet Exchange (IPX) is derived from the Xerox Network Systems (XNS) Internet Datagram Protocol (IDP). IPX and XNS have the following differences:

- IPX and XNS do not always use the same Ethernet encapsulation format.
- IPX uses Novell's proprietary Service Advertisement Protocol (SAP) to advertise special network services. File servers and print servers are examples of services that are typically advertised.
- IPX uses ticks, while XNS uses hop count as the primary metric in determining the best path to a destination.

This chapter describes how to configure Novell IPX and provides configuration examples. For a complete description of the commands mentioned in this chapter, refer to the "Novell IPX Commands" chapter in the *Access and Communication Servers Command Reference* publication. For historical background and a technical overview of Novell IPX, see the *Internetworking Technology Overview* publication.

Cisco's Implementation of Novell IPX

Cisco's implementation of Novell's IPX protocol has been certified as providing full IPX router functionality. When configured as a router, a Cisco communication server connects Ethernet and Token Ring networks, either directly or through high-speed serial lines (56 kbps to T1 speeds), X.25, or Frame Relay. At this time, the Cisco X.25 and T1 support is not compatible with Novell. This means that our communication servers must be used on both ends of T1 and X.25 circuits.

Cisco supports the IPX MIB. The IPX Accounting group represents one of the local variables we support. This group provides access to the active database that is created and maintained if IPX accounting is enabled on a communication server.

Cisco routers also support IPX Enhanced IGRP, which provides the following features:

- Automatic redistribution. IPX RIP routes are automatically redistributed into Enhanced IGRP, and Enhanced IGRP routes are automatically redistributed into RIP. If desired, you can turn off redistribution. You also can completely turn off Enhanced IGRP and IPX RIP on the router or on individual interfaces.
- Increased network width. With IPX RIP, the largest possible width of your network is 15 hops. When Enhanced IGRP is enabled, the largest possible width is 224 hops. Because the enhanced IGRP metric is large enough to support thousands of hops, the only barrier to expanding the network is the transport layer hop counter. Cisco works around this problem by incrementing the transport control field only when an IPX packet has traversed 15 routers and the next hop to the destination was learned via enhanced IGRP. When a RIP route is being used as the next hop to the destination, the transport control field is incremented as usual.

- Incremental SAP updates. Complete SAP updates are sent periodically on each interface until an Enhanced IGRP neighbor is found and thereafter only when there are changes to the SAP table. This procedure works by taking advantage of enhanced IGRP's reliable transport mechanism, which means that an Enhanced IGRP peer must be present for incremental SAPs to be sent. If no peer exists on a particular interface, periodic SAPs will be sent on that interface until a peer is found. This functionality is automatic on serial interfaces and can be configured on LAN media.

IPX Addresses

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

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

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

The following is an example of an IPX network address:

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

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

IPX Configuration Task List

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

- Enable IPX Routing
- Configure NLSP
- Control Access to IPX Networks
- Tune IPX Network Performance
- Configure IPX Enhanced IGRP
- Configure IPX Accounting
- Shut Down an IPX Network
- Configure IPX over WANs
- Monitor and Maintain the IPX Network

See the "Configuration Examples" section at the end of this chapter for configuration examples.

Enable IPX Routing

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

- Enable IPX Routing on the Communication Server
- Assign Network Numbers to Individual Interfaces

Enable IPX Routing on the Communication Server

The first step in enabling IPX routing is to enable it on the communication server. If you do not specify the node number of the communication server, the communication server uses the hardware media access control (MAC) address currently assigned to it as its node address. This is the MAC address of the first Ethernet or Token Ring interface card.

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

Task	Command
Enable IPX routing on the communication server.	**ipx routing** [*node*]

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

Assign Network Numbers to Individual Interfaces

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

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

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

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

Assign Network Numbers to Interfaces That Support a Single Network

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

Task	Command
Enable IPX routing on an interface.	**ipx network** *network* [**encapsulation** *encapsulation-type*]

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

For an example of how to enable IPX routing, see the section "Enabling IPX Routing Example" in this publication.

Assign Network Numbers to Interfaces That Support Multiple Networks

To assign network numbers to interfaces that support multiple networks, you normally use subinterfaces. A subinterface is a mechanism that allows a single physical interface to support multiple logical interfaces or networks. That is, several logical interfaces or networks can be associated with a single hardware interface. Each subinterface must use a distinct encapsulation, and the encapsulation must match the encapsulation tyhpe used by the clients and servers using the same network number. To run NLSP on multiple networks on the same physical LAN interface, you must configure subinterfaces.

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

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

Task		Command
Step 1	Specify a subinterface.	**interface** *type interface-number.subinterface-number*
Step 2	Enable IPX routing, specifying the first encapsulation type.	**ipx network** *network* **encapsulation** *encapsulation-type*

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

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

For examples of configuring multiple IPX networks on an interface, see the section “Enabling and Disabling IPX Routing on Multiple Networks Example” later in this chapter.

Table 20-1 lists the encapsulation types you can use on IEEE interfaces and shows the correspondence between the encapsulation type and the IPX frame type.

Table 20-1 Novell IPX Encapsulation Types on IEEE Interfaces

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

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, perform the following tasks in interface configuration mode:

Task		Command
Step 1	Enable IPX routing on the primary network.	**ipx network** *network* **encapsulation** *encapsulation-type*
Step 2	Enable IPX routing on a secondary network.	**ipx network** *network* **encapsulation** *encapsulation-type* **secondary**

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

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

Configure NLSP

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

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

Level 1 routers are used to 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 current NLSP specification defines only level 1 procedures, which allow operation within a routing area and routing to the nearest level 2 router only.

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 communication servers in the area, the links connecting them, the operational status of the communication servers and links, and other related parameters. For each point-to-point link, the database records the end-point communication servers and the state of the link. For each LAN, the database records which communication servers 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.

Our implementation of NLSP is based on revision 1.0 of the Novell NLSP specification, which specifies routing with a routing area (that is, Level 1 routing). Our implementation of NLSP also includes NLSP MIB variables.

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

- Adjacency—keeps track of the communication server's immediate neighbors and the operational status of the directly attached links by exchanging hello packets. Adjacencies are created upon receipt of periodic hello packets. If a link or communication server 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). Link-state packets contain lists of adjacencies. They are flooded to all other routers via a reliable flooding algorithm every time a link state changes. LSPs are refreshed every two hours. To keep the size of the link-state database reasonable, NLSP uses fictitious pseudonodes, which represent the LAN as a whole, and designated routers, which originate LSPs on behalf of the pseudonode.
- Forwarding—calculated from the adjacency and link state databases using Dijkstra's Shortest Path First (SPF) algorithm.

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

- Define an Internal Network
- Enable NLSP Routing on the Communication Server
- Configure NLSP on an Interface

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

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

For an example of enabling NLSP, see the section "Enabling and Disabling IPX Routing Protocols Examples" in this chapter.

Define an Internal Network

An internal network number is a number assigned to the communication server. In order for NLSP to operate, you must configure an internal network number for each communication server.

To enable IPX routing and define an internal network numbers, perform the following task in global configuration mode:

Task	Command
Enable IPX routing.	**ipx routing**
Define an internal network number.	**ipx internal-network** *network-number*

Enable NLSP Routing on the Communication Server

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

Task		Command
Step 1	Enable NLSP on the communication server.	**ipx router nlsp**
Step 2	Define a set of network numbers to be part of the current NLSP area.	**area-address** *address mask*

Configure NLSP on an Interface

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

Configure NLSP on a LAN Interface

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

Task		Command
Step 1	Enable IPX routing on an interface.	**ipx network** *network* [**encapsulation** *encapsulation-type*]
Step 2	Enable NLSP on the interface.	**ipx nlsp enable**

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

Task		Command
Step 1	Specify a subinterface.	**interface** *type interface-number.subinterface-number*
Step 2	Enable IPX routing, specifying the first encapsulation type.	**ipx network** *network* **encapsulation** *encapsulation-type*
Step 3	Enable NLSP on the subinterface.	**ipx nlsp enable**

Repeat these three steps for each subinterface.

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

Configure NLSP on a WAN Interface

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

Task		Command
Step 1	Specify a serial interface.	**interface serial** *number*
Step 2	Enable IPXWAN.	**ipx ipxwan** [*local-node* **unnumbered** *local-server-name retry-interval retry-limit*]
Step 3	Enable NLSP on the interface.	**ipx nlsp enable**

Configure 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 udpates on a network enabled for NLSP, perform one of the following tasks in interface configuration mode:

Task	Command
Never generate RIP periodic traffic.	**ipx nlsp rip off**
Always generate RIP periodic traffic.	**ipx nlsp rip on**
Send RIP periodic traffic only if another RIP router is sending periodic RIP traffic. (This is the default on interfaces configured for NLSP.)	**ipx nlsp rip auto**

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

Task	Command
Never generate SAP periodic traffic.	**ipx nlsp sap off**
Always generate SAP periodic traffic.	**ipx nlsp sap on**
Send SAP periodic traffic only if another SAP service is sending periodic SAP traffic. (This is the default on interfaces configured for NLSP.)	**ipx nlsp sap auto**

Configure Maximum Hop Count

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

Task	Command
Set the maximum hop count accepted from RIP update packets.	**ipx maximum-hops** *hops*

Configure the Link Delay and Throughput

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

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

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

Task	Command
Specify the link delay.	**ipx link-delay** *microseconds*

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

Task	Command
Specify the throughput.	**ipx throughput** *bits-per-second*

Configure the Metric Value

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

Task	Command
Set the metric value for an interface.	**ipx nlsp metric** *metric-number*

Configure the Priority of the System for Designated Router Election

Note In the context of this discussion, the term "designated router" refers to a communication server set up as a designated router.

NLSP elects a designated router on each LAN interface. This router creates a virtual router called a pseudonode, which generates routing information on behalf of the LAN and transmits it to the rest of the routing area. The routing information generated includes adjacencies and RIP routes. The use of a designated router significantly reduces the number of entries in the adjacency database.

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

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

Task	Command
Configure the designated router election priority.	**ipx nlsp priority** *priority-number*

Configure Default Routes

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

Task	Command
Disable default route handling.	**ipx default-route**

Unless configured otherwise, all known routes are advertised out each interface. However, you can choose to advertise only the default route if it is known. This greatly reduces the CPU overhead when routing tables are large. Note that services are not considered to be reachable via the default route

alone. A specific route to the destination network must be known before a service advertisement will be accepted. Therefore, advertise only the default route with caution if services are to be advertised via the interface.

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

Task	Command
Advertise only the RIP default route	**ipx advertise-default-route-only** *network*

Configure Transmission and Retransmission Intervals

You can configure the hello and CSNP transmission intervals, and the LSP retransmission interval.

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

Task	Command
Configure the hello transmission interval.	**ipx nlsp hello-interval** *seconds*

To configure the CSNP transmission interval, perform the following task in interface configuration mode:

Task	Command
Configure the CSNP transmission interval.	**ipx nlsp csnp-interval** *seconds*

To configure the LSP retransmission interval, perform the following task in interface configuration mode:

Task	Command
Configure the LSP retransmission interval.	**ipx nlsp retransmit-interval** *seconds*

Modify Link-State Packet (LSP) Parameters

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

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

Control Access to IPX Networks

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

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

- Standard access list—Restricts traffic based on the source network number. You can further restrict traffic by specifying a destination address and a source and destination address mask. Standard IPX access lists have numbers from 800 to 899.
- Extended access list—Restricts traffic based on the IPX protocol type. You can further restrict traffic by specifying source and destination addresses and address masks, and source and destination sockets. Extended IPX access lists have numbers from 900 to 999.
- SAP access list—Restricts traffic based on the IPX Service Advertisement Protocol (SAP) type. These lists are used for SAP filters and Get Nearest Server (GNS) response filters. Novell SAP access lists have numbers from 1000 to 1099.
- IPX NetBIOS access list—Restricts IPX NetBIOS traffic based on NetBIOS names, not numbers.

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

- Generic output filters—Control which packets are routed out an interface based on the packet's source and destination addresses and IPX protocol type.
- Routing table filters—Control which Routing Information Protocol (RIP) updates are accepted and advertised by the communication server and which communication servers the local communication server accepts RIP updates from.
- SAP filters—Control which SAP services the communication server accepts and advertises and which Get Nearest Server (GNS) response messages it sends out.
- IPX NetBIOS filters—Control incoming and outgoing IPX NetBIOS packets.
- Broadcast filters—Control which broadcast packets are forwarded.

Table 20-2 summarizes the types of filters and the commands you use to define them. Use the **show ipx interfaces** command to display the filters defined on an interface.

Table 20-2 IPX Filters

Filter Type	Command Used to Define Filter
Generic filters	
Filter outbound packets based on protocol, address and address mask, and socket.	**ipx access-group** *access-list-number*
Routing table filters	
Control which networks are added to the routing table.	**ipx input-network-filter** *access-list-number*
Control which networks are advertised in routing updates.	**ipx output-network-filter** *access-list-number*
Control the communication servers from which updates are accepted.	**ipx router-filter** *access-list-number*
SAP filters	
Filter incoming service advertisements.	**ipx input-sap-filter** *access-list-number*
Filter outgoing service advertisements.	**ipx output-sap-filter** *access-list-number*
Control the communication servers from which SAP updates are accepted.	**ipx router-sap-filter** *access-list-number*
Filter list of servers in GNS response messages.	**ipx output-gns-filter** *access-list-number*
IPX NetBIOS filters	
Filter incoming packets by node name.	**ipx netbios input-access-filter host** *name*

Filter Type	Command Used to Define Filter
Filter incoming packets by byte pattern.	**ipx netbios input-access-filter bytes** *name*
Filter outgoing packets by node name.	**ipx netbios output-access-filter host** *name*
Filter outgoing packets by byte pattern.	**ipx netbios output-access-filter bytes** *name*
Broadcast filters	
Control which broadcast packets are forwarded.	**ipx helper-list** *access-list-number*

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

- Access lists entries are scanned in the order you enter them. The first matching entry is used. To improve performance, it is recommended that you place the most commonly used entries near the beginning of the access list.
- An implicit *deny everything* entry is defined at the end of an access list unless you include an explicit *permit everything* entry at the end of the list.
- All new entries to an existing list are placed at the end of the list. You cannot add an entry to the middle of a list. This means that if you have previously included an explicit *permit everything* entry, new entries will never be scanned. The solution is to delete the access list and re-enter it with the new entries.
- Take care not to set up conditions that result in packets getting lost. One way this can happen is when a communication server or interface is configured to advertise services on a network that has access lists that deny these packets.

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

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

Create Access Lists

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

Task	Command
Create a standard IPX access list (for generic, routing, and broadcast filters).	**access-list** *access-list-number* {**deny** | **permit**} *source-network*[.*source-node*[*source-node-mask*]] *[destination-network*[.*destination-node* [*destination-node-mask*]]]

Task	Command
Create an extended IPX access list (for generic, routing, and broadcast filters).	**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*]]
Create an IPX access list for SAP filters.	**access-list** *access-list-number* {**deny** | **permit**} *network*[**.***node*] [*network-mask node-mask*] [*service-type* [*server-name*]]
Create an access list for filtering IPX NetBIOS packets by node name.	**netbios access-list host** *name* {**deny** | **permit**} *string*
Create an access list for filtering IPX NetBIOS packets by arbitrary byte pattern.	**netbios access-list bytes** *name* {**deny** | **permit**} *offset byte-pattern*

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

Create Generic Filters

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

To create generic filters, perform the following tasks:

Step 1 Create a standard or an extended access list.

Step 2 Apply a filter to an interface.

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

Task	Command
Create a standard IPX access list.	**access-list** *access-list-number* {**deny** | **permit**} *source-network*[.*source-node*[*source-node-mask*]] *[destination-network*[.*destination-node* [*destination-node-mask*]]]
Create an extended IPX access list.	**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*]]

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

Task	Command
Apply a generic filter to an interface.	**ipx access-group** *access-list-number*

For an example of creating a generic filter, see the section "IPX Network Access Example."

Create Filters for Updating the Routing Table

Routing table update filters control the entries that the communication server accepts for its routing table and the networks that it advertises in its routing updates.

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

Step 1 Create a standard or an extended access list.

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

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

Task	Command
Create a standard IPX access list.	**access-list** *access-list-number* {**deny** | **permit**} *source-network*[.*source-node*[*source-node-mask*]] *[destination-network*[.*destination-node* [*destination-node-mask*]]]
Create an extended IPX access list.	**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*]]

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

Task	Command
Control which networks are added to the routing table when IPX routing updates are received.	**ipx input-network-filter** *access-list-number*
Control which networks are advertised in routing updates sent out by the communication server.	**ipx output-network-filter** *access-list-number*
Control the communication servers from which routing updates are accepted.	**ipx router-filter** *access-list-number*

You can apply one filter to each interface.

Create SAP Filters

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

Step 1 Create a SAP access list.

Step 2 Apply one or more filters to an interface.

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

Task	Command
Create a SAP access list.	**access-list** *access-list-number* {**deny** | **permit**} *network*[.*node*] [*network.node-mask*] [*service-type* [*server-name*]]

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

Task	Command
Filter incoming service advertisements.	**ipx input-sap-filter** *access-list-number*
Filter outgoing service advertisements.	**ipx output-sap-filter** *access-list-number*
Filter service advertisements received from a particular communication server.	**ipx router-sap-filter** *access-list-number*

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

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

Create GNS Response Filters

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

Step 1 Create a SAP access list.

Step 2 Apply a GNS filter to an interface.

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

Task	Command
Create a SAP access list.	**access-list** *access-list-number* {**deny** | **permit**} *network*[.*node*] [*network.node-mask*] [*service-type* [*server-name*]]

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

Task	Command
Filter the list of servers in GNS response messages.	**ipx output-gns-filter** *access-list-number*

Create IPX NetBIOS Filters

Novell's IPX NetBIOS allows messages to be exchanged between nodes using alphanumeric names as well as node addresses. Therefore, the communication server lets you filter incoming and outgoing NetBIOS 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 packets only. They have no effect on LLC2 NetBIOS packets.

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

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

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

Step 1 Create a NetBIOS access list.

Step 2 Apply the access list to an interface.

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

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

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

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

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

Create Broadcast Message Filters

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

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

Step 1 Create an access list.

Step 2 Apply a broadcast message filter to an interface.

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

Task	Command
Create a standard IPX access list.	**access-list** *access-list-number* {**deny** | **permit**} *source-network*[.*source-node*[*source-node-mask*]] *[destination-network*[.*destination-node* [*destination-node-mask*]]]
Create an extended IPX access list.	**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*]]

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

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

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 section “Helper Facilities to Control Broadcasts Examples” later in this chapter.

Tune IPX Network Performance

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

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

- Configure the Queue Length for SAP Requests
- Configure RIP Update Packet Size
- Adjust SAP Update Timers
- Configure SAP Update Packet Size
- Set Maximum Paths
- Control Responses to GNS Requests
- Use Helper Addresses to Forward Broadcast Messages
- Control the Forwarding of Type 20 Packets
- Repair Corrupted Network Numbers

Control Novell IPX Compliance

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

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

Task	Command
Restrict the acceptance of IPX type 20 propagation packets.	**ipx type-20-input-checks**
Restrict the forwarding of IPX type 20 propagation packets.	**ipx type-20-output-checks**
Set the interpacket delay of multiple-packet routing updates sent on all interfaces.	**ipx default-output-rip-delay** *delay*
Set the interpacket delay of multiple-packet triggered routing updates sent on all interfaces.	**ipx default-triggered-rip-delay** *delay*
Set the interpacket delay of multiple-packet Service Advertisement Protocol (SAP) updates sent on all interfaces.	**ipx default-output-sap-delay** *delay*
Set the interpacket delay of multiple-packet triggered SAP updates sent on all interfaces.	**ipx default-triggered-sap-delay** *delay*

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

Task	Command
Set the tick count, which is used in the IPX Routing Information Protocol (RIP) delay field.	**ipx delay** *number*
Administratively bring down an IPX network on an interface. This removes the network from the interface.	**ipx down** *network*
Set the interpacket delay of multiple-packet routing updates sent on a single interface.	**ipx output-rip-delay** *delay*

Task	Command
Set the interpacket delay of multiple-packet triggered routing updates sent on a single interface.	**ipx triggered-rip-delay** *delay*
Set the interpacket delay of multiple-packet SAP updates sent on a single interface.	**ipx output-sap-delay** *delay*
Set the interpacket delay of multiple-packet triggered SAP updates sent on a single interface.	**ipx triggered-sap-delay** *delay*
Forward IPX type 20 propagation packets to other network segments.	**ipx type-20-propagation**

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

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

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

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

Task		Command
Step 1	Set the interpacket delay of multiple-packet routing updates sent on all interfaces to 55 ms.	**ipx default-output-rip-delay 55**
Step 2	Set the interpacket delay of multiple-packet SAP updates sent on all interfaces to 55 ms.	**ipx default-output-sap-delay 55**

Configure Static Routes

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

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

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

Task	Command
Add a static route to the routing table.	**ipx route** [*network* | **default**] *network.node*

You can configure static routes that can be overridden by dynamically learned routes. These are referred to a 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 that by default, floating-static routes are not redistributed into other dynamic protocols.

To add a floating-static route to the communication server's routing table, perform the following task in global configuration mode:

Task	Command
Add a floating-static route to the routing table.	**ipx route** *network network.node* **floating-static**

Adjust Routing Table Update Timers

You can set the interval between IPX RIP updates on a per-interface basis. You also can specify that a delay be inserted between the packets of a multiple-packet update.

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

- IPX routes are marked invalid if no routing updates are heard within three times the value of *the update interval* (3 * *interval*) and are advertised with a metric of infinity.
- IPX routes are removed from the routing table if no routing updates are heard within four times the value of *the update interval* (4 * *interval*).
- If you define an update timer for more than one interface in a communication server, the granularity of the update timer is determined by the lowest value defined for one of the interfaces in the communication server. The communication server "wakes up" at this granularity interval and sends out updates as appropriate. For more information about granularity, see the "Novell IPX Commands" chapter in the *Access and Communication Servers Command Reference* publication.

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

To adjust RIP update times on the communication server, perform one or both of the following tasks in interface configuration mode:

Task	Command
Adjust the RIP update interval.	**ipx update-time** *seconds*
Adjust the delay between multiple-packet routing updates.	**ipx output-rip-delay** *milliseconds*

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

Task	Command
Configure the interval at which a network RIP entry ages out.	**ipx rip-multiplier** *multiplier*

Configure RIP Update Packet Size

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

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

Configure Static SAP Table Entries

Servers use SAP to advertise their services via broadcast packets. Communication servers store this information in the SAP table, also known as the Server Information Table (SIT). This table is updated dynamically. You might want to explicitly add an entry to the SIT 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 communication server will not announce the static SAP entry until it relearns the route.

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

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

Configure the Queue Length for SAP Requests

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

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

Task	Command
Configure the maximum SAP queue length.	**ipx sap-queue-maximum** *number*

Adjust SAP Update Timers

You can adjust the interval at which SAP updates are sent, and you can set the delay between packets sent in multipacket SAP updates.

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

Adjusting the delay between packets sent in a multipacket SAP update is useful when the IPX network has slow IPX servers or communication servers. Setting a delay between packets in a multipacket SAP update forces our communication server interface to slow its output of SAP packets.

To modify the SAP timers used by the communication server, perform one or both of the following tasks in interface configuration mode:

Task	Command
Adjust the interval at which SAP updates are sent.	**ipx sap-interval** *minutes*
Adjust the delay between packets sent in multiple-packet SAP updates.	**ipx output-sap-delay** *milliseconds*

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

Task	Command
Configure the interval at which a network's or server's SAP enty ages out.	**ipx sap-sap-multiplier** *multiplier*

Configure SAP Update Packet Size

By default, the maximum size of SAP updates sent out 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, perform the following task in interface configuration mode:

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

Set Maximum Paths

You can set the maximum number of equal-cost, parallel paths to a destination. (Note that when paths have differing costs, the communication server chooses lower-cost routes in preference to higher-cost routes.) The communication server 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 whether or not fast switching is enabled.

The cost of a path is determined by ticks, with hop count used as a tie breaker.

Limiting the number of equal-cost paths can save memory on communication servers 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 on the communication server, perform the following task in global configuration mode:

Task	Command
Set the maximum number of equal-cost paths to a destination.	**ipx maximum-paths** *paths*

Control Responses to GNS Requests

You can set the method in which the communication server responds to SAP GNS requests, and you can set the delay time in responding to these requests. By default, the communication server responds to GNS requests. However, you can disable this.

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

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

Task	Command
Respond to GNS requests using a round-robin selection method.	**ipx gns-round-robin**
Set the delay when responding to GNS requests.	**ipx gns-response-delay** [*milliseconds*]
Disable the sending of replies to GNS queries.	**ipx gns-reply-disable**

Use Helper Addresses to Forward Broadcast Messages

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

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

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

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

Task	Command
Specify a helper address for forwarding broadcast messages.	**ipx helper-address** *network.node*

You can specify multiple helper addresses on a given interface.

For an example of using helper addresses to forward broadcast messages, refer to the section “Helper Facilities to Control Broadcasts Examples” later in this chapter.

Control the Forwarding of Type 20 Packets

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

Communication servers normally block all broadcast requests. By enabling type 20 packet propagation, IPX interfaces on the communication server may accept and forward type 20 propagation packets. Before forwarding (flooding) the packets, the communication server performs loop detection as described by the IPX Router Specification.

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

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

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

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

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

Enable the Forwarding of Type 20 Packets

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

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

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

Restrict the Acceptance of Incoming Type 20 Packets

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

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

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

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

Restrict the Forwarding of Outgoing Type 20 Packets

For outgoing type 20 propagation packets, the communication server 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 communication server to send these packets only to networks that are not routes back to the source network. (The communication server uses the current routing table to determine routes.)

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

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

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

Repair Corrupted Network Numbers

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

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

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

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

Configure IPX Enhanced IGRP

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

The convergence technology is based on research conducted at SRI International and employs an algorithm referred to as the Diffusing Update Algorithm (DUAL). This algorithm guarantees loop-free operation at every instant throughout a route computation and allows all communication servers involved in a topology change to synchronize at the same time. Communication servers 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.

Cisco's Implementation of IPX Enhanced IGRP

IPX Enhanced IGRP provides the following features:

- Automatic redistribution. IPX RIP routes are automatically redistributed into Enhanced IGRP, and IPX Enhanced IGRP routes are automatically redistributed into RIP. If desired, you can turn off redistribution. You also can completely turn off IPX Enhanced IGRP and IPX RIP on the router or on individual interfaces.
- Increased network width. With IPX RIP, the largest possible width of your network is 15 hops. When IPX Enhanced IGRP is enabled, the largest possible width is 224 hops. Because the Enhanced IGRP metric is large enough to support thousands of hops, the only barrier to expanding the network is the transport layer hop counter. Cisco works around this problem by

incrementing the transport control field only when an IPX packet has traversed 15 routers and the next hop to the destination was learned via Enhanced IGRP. When a RIP route is being used as the next hop to the destination, the transport control field is incremented as usual.

- Incremental SAP updates. Complete SAP updates are sent periodically on each interface until an IPX Enhanced IGRP neighbor is found and thereafter only when there are changes to the SAP table. This procedure works by taking advantage of Enhanced IGRP's reliable transport mechanism, which means that an IPX Enhanced IGRP peer must be present for incremental SAPs to be sent. If no peer exists on a particular interface, periodic SAPs will be sent on that interface until a peer is found. This functionality is automatic on serial interfaces and can be configured on LAN media.

IPX Enhanced IGRP Configuration Task List

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

- Enable IPX Enhanced IGRP
- Configure Miscellaneous Parameters
- Monitor IPX Enhanced IGRP on an IPX Network

See the "Configuration Examples" section at the end of this chapter for configuration examples.

Enable IPX Enhanced IGRP

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

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

For an example of how to enable IPX Enhanced IGRP, see the section "Enabling and Disabling IPX Routing Protocols Examples."

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

Configure Miscellaneous Parameters

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

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

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

Redistribute Routing Information

By default, the router redistributes IPX RIP routes into IPX Enhanced IGRP, and vice versa. When routes are redistributed, a RIP route to a destination with a hop count of 1 is always preferred over an Enhanced IGRP route with a hop count of 1. This ensures that the router always believes a Novell IPX server over a Cisco router for internal IPX networks. The only exception to this rule is if both the RIP and Enhanced IGRP updates were received from the same router. In this case, and in the case of all other RIP metrics (2 through 15), the Enhanced IGRP route always is preferred over the RIP route when the hop counts are the same.

Internal Enhanced IGRP routes are always preferred over external Enhanced IGRP routes. This means that if there are two Enhanced IGRP paths to a destination, the path that originated within the Enhanced IGRP autonomous system will always be preferred over the Enhanced IGRP path that originated from outside of the autonomous system, regardless of the metric. Redistributed RIP routes are always advertised in Enhanced IGRP as external.

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

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

Adjust the Interval between Hello Packets and the Hold Time

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

Routers periodically send hello packets to each other to dynamically learn of other routers on their directly attached networks. Routers use this information to discover who their neighbors are and to discover when their neighbors become unreachable or inoperative. By default, hello packets are sent every 5 seconds.

You can configure the hold time, in seconds, on a specified interface for the IPX 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.

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

Task	Command
Set the interval between hello packets.	**ipx hello-interval eigrp** *autonomous-system-number 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 do this, perform the following task in interface configuration mode:

Task	Command
Set the hold time.	**ipx hold-time eigrp** *autonomous-system-number seconds*

Note Do not adjust the hold time without advising technical support.

Disable Split Horizon

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

By default, split horizon is enabled on all interfaces.

Split horizon blocks information about routes from being advertised by a router 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 may wish to disable split horizon.

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

Task	Command
Disable split horizon.	**no ipx split-horizon eigrp** *autonomous-system-number*

Control SAP Updates

If IPX Enhanced IGRP peers are found on an interface, you can configure the communication server 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 communication server sends SAP updates only when the SAP table changes. On Ethernet, Token Ring, and FDDI interfaces, by default, the communication server sends SAP updates periodically. To reduce the amount of bandwidth required to send SAP updates, you might want to disable the periodic sending of SAP updates on LAN interfaces. Do this only when all nodes out this interface are Enhanced IGRP peers; otherwise, loss of SAP information on the other nodes will result.

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

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

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

Task	Command
Send SAP updates periodically.	**no ipx sap-incremental eigrp** *autonomous-system-number*

For an example of how to configure SAP updates, see the section "Enhanced IGRP SAP Update Examples" in this publication.

Control the Advertising of Routes in Routing Updates

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

Task	Command
Control the advertising of routes in routing updates.	**distribute-list** *access-list-number* **out** [*interface-name* | *routing-process*]

Control the Processing of Routing Updates

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

Task	Command
Control which incoming route updates are processes.	**distribute-list** *access-list-number* **in** [*interface-name*]

Query the Backup Server

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

By default, the router queries its own copy of each Enhanced IGRP neighbor's backup server table every 15 seconds. To change this interval, perform the following global configuration task:

Task	Command
Specify the minimum period of time between successive queries of a neighbor's backup server table.	**ipx backup-server-query-interval** *interval*

Log Enhanced IGRP Neighbor Adjacency Changes

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

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

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

Configure the Percentage of Link Bandwidth Used by Enhanced IGRP

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

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

Task	Command
Configure the percentage of bandwidth that may be used by Enhanced IGRP on an interface.	**ipx eigrp-bandwidth-percent** *percent*

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

Monitor IPX Enhanced IGRP on an IPX Network

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

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

Configure IPX Accounting

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

IPX accounting statistics are accurate even if IPX fast switching is enabled or if IPX access lists are being used. However, IPX accounting does not keep statistics if autonomous switching is enabled.

The communication server software maintains two accounting databases: an active database and a checkpointed database.

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

Task	Command
Enable IPX accounting.	**ipx accounting**

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

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

Shut Down an IPX Network

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

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

Task	Command
Shut down an IPX network but have the network still exist in the configuration.	**ipx down** *network*

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

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

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

For an example of shutting down an IPX network, see the section “Enabling IPX Routing Example” later in this chapter.

Configure IPX over WANs

You can configure IPX over dial-on-demand routing (DDR), Frame Relay, Point-to-Point Protocol (PPP), Switched Multimegabit Data Service (SMDS), and X.25 networks. To do this, you configure the appropriate address mappings as described in the appropriate chapter of this publication. You can also configure IPX over Point-to-Point Protocol (PPP); address maps are not necessary for this protocol. You can also fast switch IPX over serial interfaces configured for Frame Relay.

Additionally, you can configure the IPXWAN protocol.

Configure IPX over DDR

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

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

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

Configure the IPXWAN Protocol

Our communication servers support the IPXWAN protocol, as defined in RFC 1362. IPXWAN allows a communication server that is running IPX routing to connect via a serial link to another communication server, 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 communication servers at both ends of the serial link must be our communication servers.

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

Task		Command
Step 1	Ensure that you have not configured an IPX network number on the interface.	**no ipx network**
Step 2	Enable PPP.	**encapsulation ppp**[1]
Step 3	Enable IPXWAN.	**ipx ipxwan** [*local-node* {*network-number* | **unnumbered**} *local-server-name retry-interval retry-limit*]

Task		Command
Step 4	Optionally, define how to handle IPXWAN when a serial link fails.	**ipx ipxwan error** [**shutdown** | **reset** | **resume**]
Step 5	Optionally, enable static routing with IPXWAN.	**ipx ipxwan static**

1. This command is documented in the "Interface Commands" chapter of the *Access and Communication Servers Command Reference* publication.

Monitor and Maintain the IPX Network

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

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

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

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

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

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

Configuration Examples

The following sections provide IPX configuration examples:

- Enabling IPX Routing Example
- Enabling and Disabling IPX Routing on Multiple Networks Example
- Enabling and Disabling IPX Routing Protocols Examples
- Enabling IPX Enhanced IGRP Example
- IPX Enhanced IGRP Bandwidth Configuration Example
- Enabling IPX over a WAN Interface Example
- IPX over DDR Example
- IPX Network Access Example
- SAP Input Filter Example
- SAP Output Filter Example
- Enhanced IGRP SAP Update Examples
- IPX NetBIOS Filter Examples
- Helper Facilities to Control Broadcasts Examples
- IPX Accounting Example

Enabling IPX Routing Example

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

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

Enabling and Disabling IPX Routing on Multiple Networks Example

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

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

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

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

```
interface ethernet 0.3
shutdown
```

To bring down network 1, use the following commands:

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

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

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

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

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

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

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

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

To bring down network 1, use the following command:

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

To bring network back up, use the following command:

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

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

```
no ipx network

no ipx network 1
```

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

```
no ipx network 2
```

Enabling and Disabling IPX Routing Protocols Examples

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

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

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

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

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

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

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

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

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

```
ipx nlsp enable
!
interface ethernet 1
ipx network e1 encapsulation sap
ipx nlsp enable
```

Enabling IPX Enhanced IGRP Example

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

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

IPX Enhanced IGRP Bandwidth Configuration Example

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

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

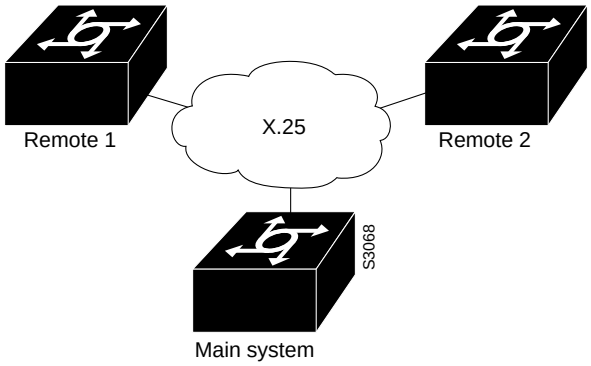
In the following example, the bandwidth of a 56 kbps circuit has been configured to be 20 kbps for routing policy reasons. The Enhanced IGRP process 109 is configured to use a maximum of 200 percent (or 40 kbps) of the circuit.

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

Enabling IPX over a WAN Interface Example

When you configure the communication server to transport IPX packets over a serial interface that is running a WAN protocol such as X.25 or PPP, you specify how the packet will be encapsulated for transport. This encapsulation is not the same as the encapsulation used on an IPX LAN interface. Figure 20-1 illustrates IPX over a WAN interface.

Figure 20-1 IPX over a WAN Interface



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

Configuration for Main System

```
hostname Main
!
no ip routing
novell routing 0000.0c17.d726
!
interface Ethernet0
no ip address
Novell network 100
media-type 10BaseT
!
interface Serial0
no ip address
shutdown
!
interface Serial1
no ip address
encapsulation x25
x25 address 33333
x25 htc 28
!
interface Serial1.1 point-to-point
no ip address
novell network 2
x25 map novell 2.0000.0c03.a4ad 11111 BROADCAST
!
interface Serial1.2 point-to-point
no ip address
novell network 3
x25 map novell 3.0000.0c07.5e26 55555 BROADCAST
```

Configuration for Remote 1

```
hostname Remote1
!
no ip routing
novell routing 0000.0c03.a4ad
!
interface Ethernet0
no ip address
novell network 1
```

```
!
interface Serial0
no ip address
encapsulation x25
novell network 2
x25 address 11111
x25 htc 28
x25 map novell 2.0000.0c17.d726 33333 BROADCAST
```

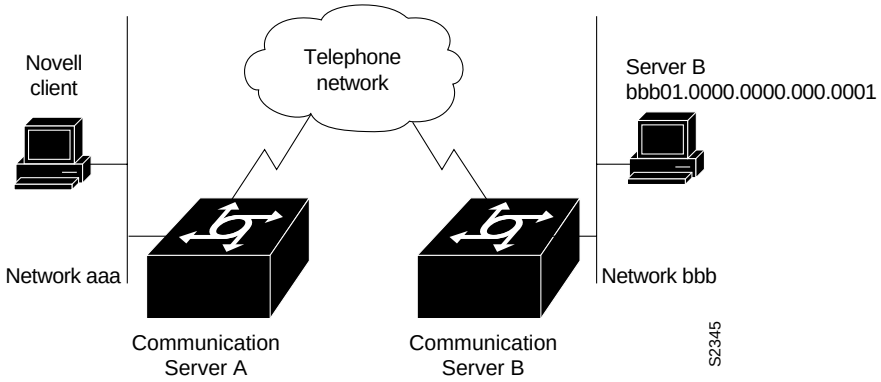
Configuration for Remote 2

```
hostname Remote2
!
no ip routing
novell routing 0000.0c07.5e26
!
interface Ethernet0
no ip address
novell network 4
media-type 10BaseT
!
interface Serial0
no ip address
shutdown
!
interface Serial1
no ip address
encapsulation x25
novell network 3
x25 address 55555
x25 htc 28
x25 map novell 3.0000.0c17.d726 33333 BROADCAST
```

IPX over DDR Example

In the configuration shown in Figure 20-2, an IPX client is separated from its server by a DDR telephone line.

Figure 20-2 IPX over DDR Configuration



Routing and service information is sent every 60 seconds. The output RIP and SAP filters defined in this example filter these updates, preventing them from being sent between Communication Servers A and B. If you were to forward these packets, the two communication servers would each have to telephone the other every 60 seconds. On a serial link that charges based on the number of packets transmitted, this is generally not desirable. This might not be an issue on a dedicated serial line.

Once the server and client have established contact, the server will send keepalive (watchdog) packets regularly. The purpose of these packets is to ensure that the connection between the server and the client is still functional; these packets contain no other information. Servers send watchdog packets approximately every five minutes. If you were to allow Communication Server B to forward the server's keepalive packets to Communication Server A and the client, Communication Server B would have to telephone Communication Server A every five minutes just to send these packets. Again, on a serial link that charges based on the number of packets transmitted, this is generally not desirable. Instead of having Communication Server B telephone Communication Server A only to send keepalive packets, you can enable watchdog spoofing on Communication Server B. This way, when the server connected to this communication server sends keepalive packets, Communication Server B will respond on behalf of the remote client (the client connected to Communication Server A).

Configuration for Communication Server A

```
access-list 1000 permit -1 7
access-list 1000 deny -1
!
!configure the communication server to which the client is connected
ipx routing 0000.0c00.59e8
ipx sap 4 SERVER-B BBB01.0000.0000.0001.4212
!
interface ethernet 0
ipx network aaa
!
interface serial 0
no keepalive
dialer in-band
dialer string 8986
ipx network ccc
pulse-time 1
dialer-group 1
ipx output-sap-filter 1000
!
ipx route bbb ccc.0000.0c01.d877
ipx route bbb01 ccc.0000.0c01.d877
!
access-list 800 permit ffffffff bbb01.0000.0000.0001
access-list 800 deny -1
access-list 1000 permit ffffffff bbb01.0000.0000.0001
access-list 1000 deny -1
dialer-list 1 list 800
```

Configuration for Communication Server B

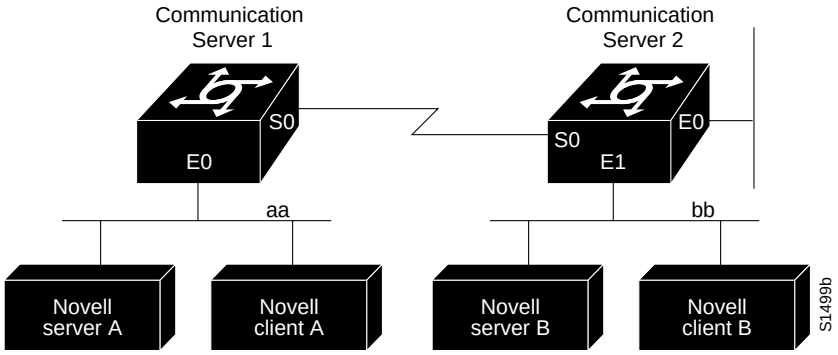
```
!configure the communication server to which the server is attached
ipx routing 0000.0x01.d877
!
interface ethernet 0
ipx network bbb
!
interface serial 1
no ip address
bandwidth 56
```

```
no keepalive
ipx output-sap-filter 1000
dialer in-band
ipx network bbb
pulse-time 1
no ipx route-cache
no ipx-route-cache cbus
!enable watchdog spoofing on the server's communication server
ipx watchdog-spoof
!
ipx route aaa ccc.0000.0c00.59e8
access-list 1000 permit 4 bbb01.0000.0001
access-list 1000 deny -1
```

IPX Network Access Example

Using access lists to manage traffic routing can be a powerful tool in overall network control. However, it requires a certain amount of planning and the appropriate application of several related commands. Figure 20-3 illustrates a network featuring two communication servers on two network segments.

Figure 20-3 Novell IPX Servers Requiring Access Control



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

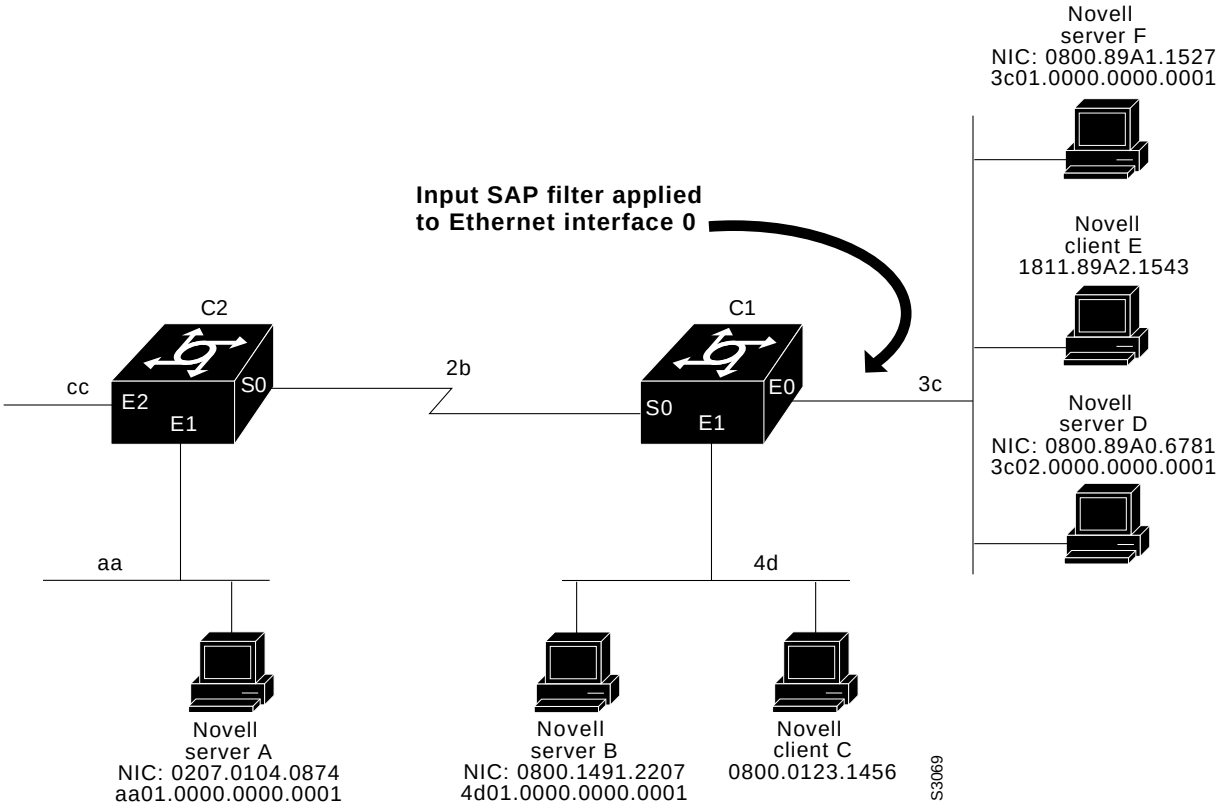
You would configure serial interface 0 on communication server 2 with the following commands:

```
ipx routing
access-list 800 deny aa bb
access-list 800 permit -1 -1
interface serial 0
ipx network bb
ipx access-group 800
```

SAP Input Filter Example

SAP input filters allow a communication server to determine whether or not to accept information about a service. Communication server CS1, illustrated in Figure 20-4, will not accept and, consequently not advertise, any information about Novell server F. However, CS1 will accept information about all other servers on the network 3c. CS2 receives information about servers D and B.

Figure 20-4 SAP Input Filter



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

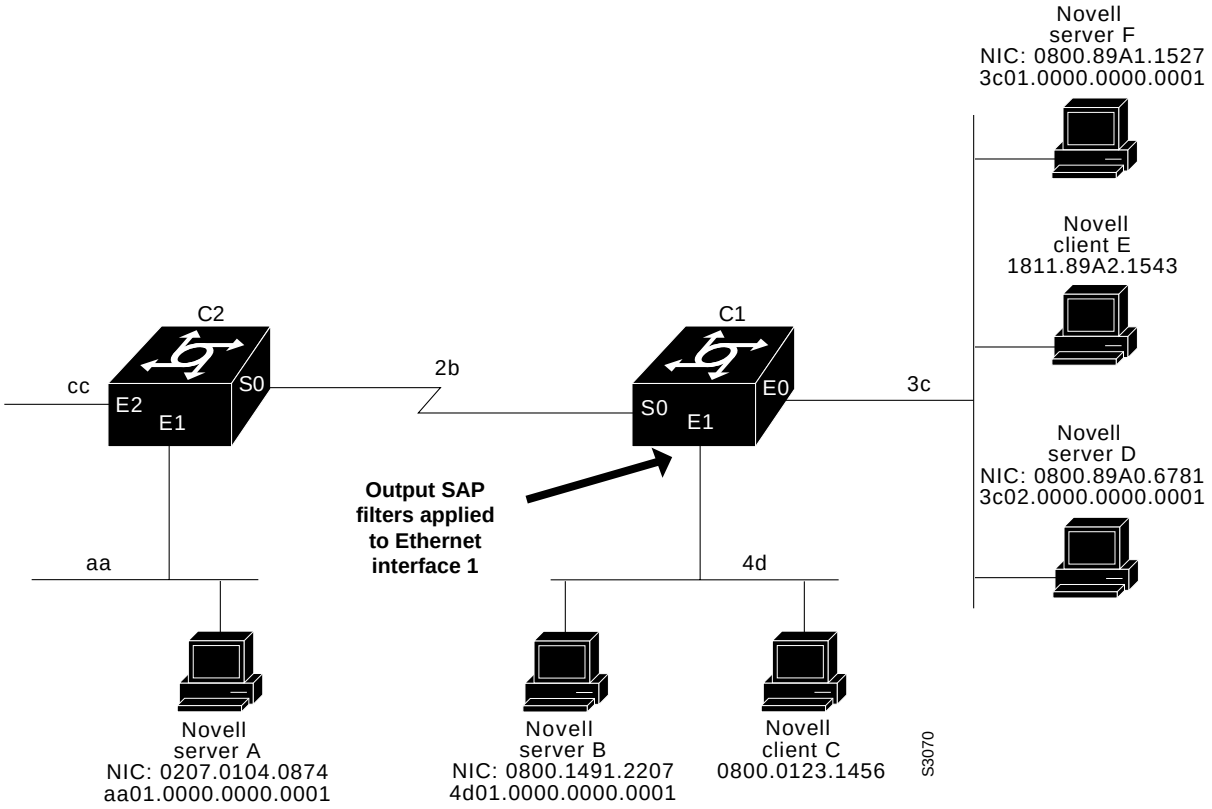
```
access-list 1000 deny 3c01.0000.0000.0001
access-list 1000 permit -1
interface ethernet 0
ipx network 3c
ipx input-sap-filter 1000
interface ethernet 1
ipx network 4d
interface serial 0
ipx network 2b
```

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

SAP Output Filter Example

SAP output filters are applied prior to the communication server sending information out a specific interface. In the example that follows, communication server C1 (illustrated in Figure 20-5) is prevented from advertising information about Novell server A out Ethernet interface 1, but can advertise server A on network 3c.

Figure 20-5 SAP Output Filter



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

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

Enhanced IGRP SAP Update Examples

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

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

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

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

IPX NetBIOS Filter Examples

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

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

```
ipx network 305 encapsulation SAP
!
interface serial 0
 no ip address
 shutdown
!
interface serial 1
 no ip address
 no keepalive
 ipx network 600
 ipx output-rip-delay 100
 ipx triggered-rip-delay 60
 ipx output-sap-delay 100
 ipx triggered-sap-delay 60
 ipx type-20-propagation
```

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

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

```
ipx type-20-propagation
ipx netbios input-access-filter bytes finigan
```

Helper Facilities to Control Broadcasts Examples

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

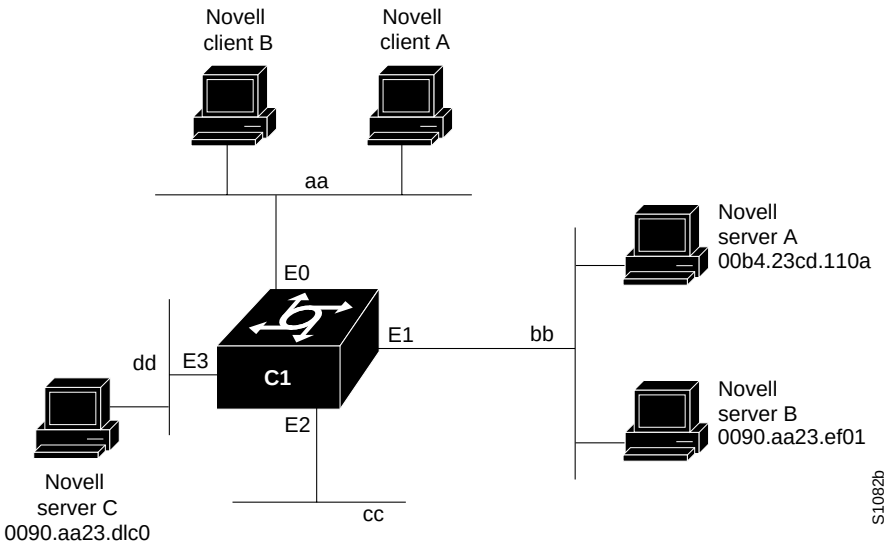
Forwarding to an Address Example

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

Figure 20-6 shows communication server C1 connected to several Ethernet interfaces. In this environment, all IPX clients are attached to segment aa, while all servers are attached to segments bb and dd. In controlling broadcasts, the following conditions are to be applied:

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

Figure 20-6 IPX Clients Requiring Server Access through a Communication Server



The following example configures the communication server shown in Figure 20-6. The first line permits broadcast traffic of type 2 from network aa. The interface and network commands configure each specific interface. The **ipx helper-address** commands permit broadcast forwarding from network aa to bb and from network aa to dd. The helper list allows type 2 broadcasts to be forwarded. (Note that type 2 broadcasts are chosen as an example only. The actual type to use depends on the application.) The **ipx type-20-propagation** command acts as a specific permission to allow type 20 broadcasts to be forwarded between networks aa and dd is also required.

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

This configuration means that any network that is downstream from network aa (for example, some arbitrary network aa1) will not be able to broadcast (type 2) to network bb through the communication server C1 unless the communication servers partitioning networks aa and aa1 are configured to forward these broadcasts with a series of configuration entries analogous to the example provided for Figure 20-6. These entries must be applied to the input interface and be set to forward broadcasts between directly connected networks. In this way, such traffic can be passed along in a directed manner from network to network. A similar situation exists for type 20 packets.

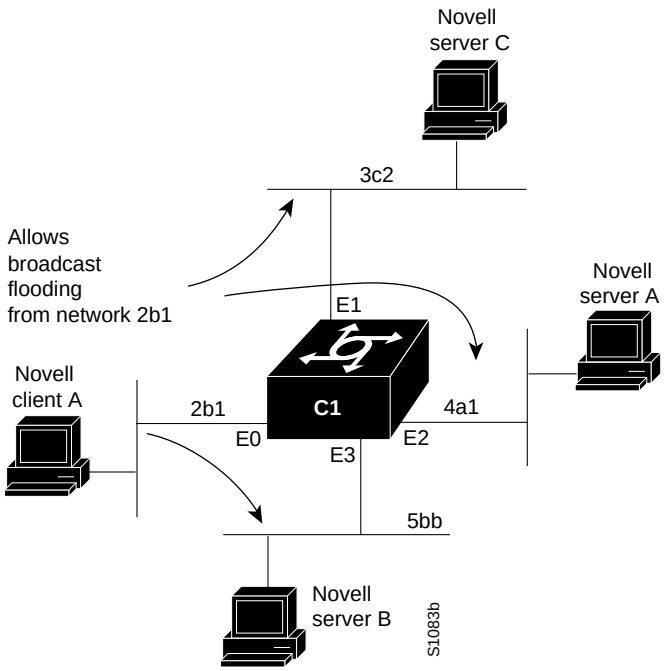
The following example rewrites the **ipx helper-address** interface configuration command line to direct broadcasts to server A:

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

Forwarding to All Networks Example

In some networks, it might be necessary to allow client nodes to broadcast to servers on multiple networks. If you configure your communication server to forward broadcasts to all attached networks, you are flooding the interfaces. In the environment illustrated in Figure 20-7, client nodes on network 2b1 must obtain services from IPX servers on networks 3c2, 4a1, and 5bb through communication server C1. To support this requirement, use the flooding address (-1.ffff.ffff.ffff) in your **ipx helper-address** interface configuration command specifications.

Figure 20-7 Type 2 Broadcast Flooding



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

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

All-Nets Flooded Broadcast Example

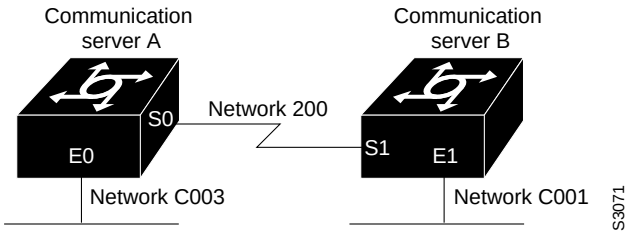
The following example configures all-nets flooding on an interface. As a result of this configuration, Ethernet interface 0 will forward all broadcast messages (except type 20) to all the networks it knows how to reach. This flooding of broadcast messages might overwhelm these networks with so much broadcast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

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

IPX Accounting Example

The following example configures two Ethernet network segments that are connected via a serial link. (See Figure 20-8.) On communication server A, IPX accounting is enabled on both the input and output interfaces (that is, on Ethernet interface 0 and serial interface 0). This means that statistics are gathered for traffic traveling in both directions (that is, out to the Ethernet network and out the serial link). However, on communication server B, IPX accounting is enabled only on the serial interface and not on the Ethernet interface. This means that statistics are gathered only for traffic that passes out the communication server on the serial link.

Figure 20-8 IPX Accounting Example



Configuration for Communication Server A

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

Configuration for Communication Server B

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