Chapter 36

Novell IPX

Introduction

This chapter describes the main features of Novell's IPX protocol, support for IPX on the router, and how to configure and operate the router to route IPX over a wide area network.

The IPX module provides the functionality of Novell's External bridge but it is actually a router in the generally accepted sense. The router also provides dial-on-demand IPX routing, using ISDN to connect LANs together when there is traffic to be transferred between the LANs. Static routes and services can be defined that are not deleted by the normal ageing mechanisms. The router spoofs responses to server requests to prevent the server from disconnecting sessions that are inactive because the ISDN call has been (temporarily) disconnected.

Some interface and port types mentioned in this chapter may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference.

The IPX Protocol

IPX (*Internetwork Packet Exchange*) is a connectionless, datagram protocol based on Xerox's XNS Internet Datagram Protocol. In addition to the basic XNS transport mechanism, IPX adds RIP (*Routing Information Protocol*) to advertise routing information, SAP (*Service Advertising Protocol*) to advertise network services, NCP (*Network Core Protocol*) to handle packet acknowledgement and connection control and SPX (*Sequenced Packet eXchange*) to provide a connection-oriented transport mechanism. These protocols use IPX as the transport mechanism.

In a Novell network, client PCs (workstations) communicate with devices that offer a *service*. File servers and print servers are two examples of services. It is not necessary to have a file server on every LAN provided that a router is used to connect file servers with workstations on remote LANs.

Like TCP/IP, IPX uses the concept of sockets to handle intranode addressing. Sockets allow different processes running within a workstation to communicate and identify themselves to IPX. External processes also use sockets to ensure that packet data is passed to the correct internal process. For instance NCP uses socket 0x451 when a request is issued from a workstation.

Addressing in a Novell Network

Novell stations must be assigned a single unique 48-bit address. This address is normally the MAC address from one of the LAN interfaces. An individual station can be connected to several physical networks but always has only one station address.

Each network must have a unique 32-bit number. There is no organisation responsible for maintaining and assigning Novell network numbers. It is up to the network administrator to ensure that each network in the internetwork has a unique network number. A number of schemes are used for assigning network numbers, including:

- Using telephone numbers.

  Use the telephone number of the office or building where the network is physically located, or the telephone number of a local support person responsible for the network. If the number is less than 8 digits long, add zeros to the left of the number. For example, the network in a laboratory, which has the telephone extension 8631 could be assigned the network number 00008631. The advantage of this method is that contact information is ready to hand if there is a network problem.

- Using IP subnet addresses.

  If the internetwork also runs IP then each network should have an administratively assigned IP subnet address which is guaranteed to be unique. A unique Novell network number can be generated by converting each of the four numbers in the IP subnet address to hexadecimal numbers and concatenating them. For example, the IP subnet address 192.168.35.32 (mask is 255.255.255.224) converted to hexadecimal is c0.a8.23.20 so the Novell network number is c0a82320. To determine the internal network number for a Netware 386 server, add one to the network number of the LAN to which the server is connected. For example, a server connected to network c0e723c0 would be assigned an internal network number of c0e723c1. This is guaranteed to be unique because no valid IP subnet address generates an odd hexadecimal number.

It is also normal to give file servers a textual name, so a full host designation looks like:

```
[00000012:008048804D32] ENGINEERING
```

In this example the host name is ENGINEERING, its station (MAC) address is 008048804d32 and its network number is 12.

Network addresses must be set for all devices which are capable of transmitting RIP or SAP packets, including routers and file servers. Each workstation does not need to have an assigned network number. It already has a unique host number by virtue of its MAC address. For details on how to change the network address on a file server consult the Novell documentation supplied with Netware.

Interfaces and Circuits

The terms *interface* and *circuit* have particular and slightly different meanings in the context of the router's implementation of IPX. The term interface has the same meaning for IPX as it does for other routing modules, referring to the underlying physical and data link layer transport mechanism for IPX. Interfaces supported by the IPX module include Point-to-Point, Frame Relay, and Ethernet (eth and vlan). A *circuit* is a logical connection over an interface, similar to an X.25 permanent virtual circuit (PVC) or a Frame Relay Data Link Connection (DLC). Some interfaces support multiple IPX circuits; other interfaces support a single IPX circuit (Table 36-1 on page 36-5).

Table 36-1: Combinations of circuits and interfaces supported by the router's implementation of IPX

Interface	Number of Circuits	Distinguishing Parameter
Ethernet	1 to 4	1 circuit per Ethernet frame type
Point-to-Point	1	-
Frame Relay	Many	1 circuit per DLCI
X.25	Many	1 circuit per channel number

The circuit is the basic unit of connectivity when configuring IPX, and includes as part of its definition the interface over which the circuit is carried.

Routing

Within a Novell network, routers exchange routing information using Novell's Routing Information Protocol (RIP).

The Novell RIP protocol is different from the TCP/IP Routing Information Protocol also called RIP. The two are similar in functionality but are not compatible in any other sense.

RIP is a distance vector protocol which uses a *hop count* metric. Like IPX, RIP is also derived from Xerox's XNS. An extra field was added by Novell to record the time that packets take to transit a path through the network. This improves the ability of the router to make a decision as to which route to choose when several choices have the same hop count. The extra field means that RIP from a Novell internet cannot be exchanged directly with an XNS-based network.

When IPX data that is addressed to a particular network is received, the routing table is scanned to find a possible route to that network. The data is then forwarded along that route by sending it to the router specified by the next hop. The table is maintained dynamically by exchanging information with other routers.

The command:

```
show ipx route
```

displays the current contents of the routing table. Networks connected directly to the router (e.g. an Ethernet LAN connection to interface eth0) have the *NextHop* field set to "Local". For other networks, the *NextHop* field contains the network and station address of the next router on the path to the network. This means that workstations on local networks which require access to services on remote networks must use this router to route all packets.

Each workstation determines the best route to a particular network by broadcasting a *RouteRequest* packet. A *RouteRequest* packet is a RIP packet with the Hop and Tick fields set to null (0). A router responds by supplying the fastest route. In its response packet the router includes its own station address. The workstation uses this MAC address as the destination field in packets that are addressed to the remote network. The remote station address is placed in the IPX destination header so that the router can forward the packet.

To maintain the routing table, each router in a Novell internet exchanges information with its nearest neighbours. The router does this by sending periodic broadcasts down its circuits. This is typically done every 60 seconds on high speed media such as Ethernet, but is normally much less frequent over slower serial links (typically 600 seconds). The time (in seconds) between RIP broadcasts can be specified when a circuit is added to the IPX module:

```
add ipx circuit=circuit interface=interface network=network
   riptimer=timer
```

The interval may be changed subsequently, using the command:

```
set ipx circuit=circuit riptimer=timer
```

All file servers on an Ethernet segment require the same RIP timer value. This should be left at the default of 60 seconds unless there is a good reason for changing it.

IPX sends the whole routing table when the timer triggers, however, deltas (i.e. changes from previous broadcasts) are sent immediately.

A circuit is added to the IPX module using the command:

```
add ipx circuit=circuit interface=interface network=network
```

A circuit is deleted from the IPX module using the command:

```
delete ipx circuit=circuit
```

The command:

```
show ipx circuit=circuit
```

displays the current configuration of an IPX circuit.

The router uses the split-horizon algorithm (sometimes called the *Best Information Algorithm* or *BIA*) to determine whether a broadcast should be sent to a particular network and what the broadcast should contain. The split-horizon algorithm has three rules:

1. RIP broadcasts sent to network *N* should not include any information about other networks that was received via network *N*.
2. RIP broadcasts sent to network *N* should not include any information about network *N*.
3. When equally good routes exist to network *N*, no information about network *N* should be included in broadcasts sent to any network from which information was received about one of the equal routes.

When a router is first powered up it places the directly connected network numbers into the routing table. It then uses the split-horizon algorithm to broadcast routing information to other routers on the directly connected circuits. It then broadcasts a request to each directly connected circuit to solicit information from other routers about other circuits. The router uses the information gained to build and maintain its routing table.

Service Advertisement

The Service Advertising Protocol (SAP) is used to propagate service information around a Novell internet. Strictly speaking a workstation client does not use SAP information directly. SAP is intended as a mechanism for keeping file servers and routers informed of the services which reside on the Novell internet. SAP is distributed in the same way as RIP. When a circuit is assigned to the router, the interval between SAP broadcasts can be defined using:

```
add ipx circuit=circuit interface=interface network=network
   saptimer=timer
```

The interval may be changed subsequently, using the command:

```
set ipx circuit=circuit saptimer=timer
```

All file servers on an Ethernet segment require the same SAP timer value. This should be left at the default of 60 seconds unless there is a good reason for changing it.

The router builds and maintains a service table listing the services, such as file servers and print servers, available on the Novell internetwork. The command:

```
show ipx service
```

displays the contents of the service table. The display is similar to that produced when the Novell SLIST command is issued on a workstation. Some services have two entries. For example, a file server may be listed as both *RConsole* and *File server*. This simply means that the file server is sending service information for two different file server service objects.

Novell introduced the concept of an internal network with Netware 386, to allow for addressing internal services. This means that Netware 386-based servers are distinguished by having a station address of 1, whereas a Netware 286-based server has a MAC address.

By convention, Version 2 Novell servers use the MAC address of the internal network interface card assigned as card A, as the station address of the server.

Within a Netware internetwork, SAP is used by workstations to get the address of a server. The workstation broadcasts a *GetNearestServer* request. All routers on the workstation's network segment which have information about the nearest server respond to this request by supplying the server's name, full Novell internetwork address and hop count. The workstation NETx.COM shell uses this to establish an initial connection to a server, so that it can use the LOGIN and ATTACH utilities (and *preferred server* option) to proceed further. Once the initial connection is made, the workstation can use RIP to establish a route to remote servers.

Traffic Filters

Traffic filters are used to control the flow of IPX packets through a network by providing a mechanism to determine whether to process IPX packets received from or destined for a specific network and/or station. The IPX module supports *inclusion* and *exclusion* lists. If there are IPX addresses on the exclusion list then IPX packets from those sources are discarded by the router. Hence packets from these excluded sources are not forwarded to other networks, or processed any further by the router itself. If there are network addresses on the inclusion list, then only packets from those sources are routed and all others are discarded. The addresses can refer to individual stations or entire networks.

The inclusion list is checked first. If the packet passes this test then the exclusion list is checked. The packet is routed when it passes both tests. This method lets an inclusion be set for an entire network but individual stations on that network may still be excluded. To set inclusions or exclusions use the commands:

```
add ipx exclusion
add ipx inclusion
```

You cannot use wildcard characters in traffic filters. Network and station addresses must be literal addresses, expressed in hexadecimal, although leading zeros may be omitted.

To delete inclusions or exclusions use the commands:

```
delete ipx exclusion
delete ipx inclusion
```

The inclusion or exclusion lists can be displayed using the commands:

```
show ipx exclusion
show ipx inclusion
```

RIP and SAP Filters

RIP and SAP filters provide a mechanism for finer control over IPX traffic in a network by filtering the entries added to the router's IPX route and service tables, and the entries in RIP and SAP broadcasts generated by the router. When an entry in a RIP or SAP packet matches one of the patterns in a filter, the filter determines whether the entry is discarded or processed. Similarly, when an entry in the router's IPX route or service table matches one of the patterns in a filter, the filter determines whether the entry appears in RIP and SAP broadcasts generated by the router. Filtering may be configured on a global or per-circuit basis.

A *filter* is a list of patterns. A *pattern* consists of:

- An action, either *inclusion* or *exclusion*. Inclusion is the action of allowing the IPX packet to be processed further and forwarded. Exclusion is the action of discarding the IPX packet.
- A *network* and *station* number to compare against the source address in an IPX packet; *or*
- A *network number* to compare against route entries in a RIP packet; *or*
- A *service name* and/or *service type* to compare against service advertisement entries in a SAP packet.

The last pattern in a filter is an implicit *match all* pattern with an exclusion action.

A linear search is performed on the filter. Searching stops at the first match found, so the order of patterns is important. Specific patterns should always appear before general patterns.

Wildcard Expressions

Patterns may contain wildcard expressions that enable sophisticated filters to be designed. Both numeric and string wildcard expressions are supported.

A numeric wildcard expression may contain the hexadecimal numeral characters "0123456789ABCDEF" and the wildcard characters "*", "%", "[]", "^" and "-" (Table 36-2 on page 36-9). If a given numeric string does not contain any wildcard characters, it is treated as a right justified number. Combinations of numeric wildcard expressions can be built up by separating the expressions with commas, for example "456BF%%%,34[4-79]*,36*,000000045*".

Table 36-2: Wildcard characters for numeric expression matching

Character	Function
*	Matches anything to the right of it. This assumes that the number is left justified. Zeroes can be used to left pad the number. For example, "000CF*" is legal but "345*89" is illegal.
%	Matches a single numeral. This can be used to provide right justified matching by using "%"s to fill the right hand side of the number; that is, "345BC5%%%%" matches "345BC54560" and "345BC5EEEE", but not "345BC5EE".

Table 36-2: Wildcard characters for numeric expression matching (Continued)

Character	Function
[]	Matches any of the characters inside the "[]", at the position it occurs. A range of consecutive numerals can be specified using the "-" character. For example, "4-7" matches the characters "4", "5", "6" and "7". If there are no characters inside the "[]", it is equivalent to a "%".

A string wildcard expression may contain any character, including the wildcard characters "*", "%", "[]", "^", "-" and "<Character" (Table 36-3 on page 36-10). Expression matching is case sensitive. Any of the wildcard characters can be used in the string to match the literal character by preceding the wildcard character with the escape character "<Character". For example, "?" is interpret as the literal "?" and "\" is treated as the literal "<Character". If a given string does not contain any wildcard characters, it is treated as an ASCII string.

Combinations of string wildcard expressions can be built up by separating the expressions with commas, for example "FRANK,FURTERS" expands to "FRANK" and "FURTERS".

Table 36-3: Wildcard characters for ASCII string expression matching

Character	Function
*	Matches any number of characters.
%	Matches a single character.
[]	Matches any of the characters inside the "[]", at the position it occurs. If the first character is a "^" the expression matches any character except the characters inside the "[]". A range of consecutive characters can be specified using the "-" character. For example, "a-z" matches any of the characters "abcdefghijklmnopqrstuvwxyz" but not any of the characters "ABCDEFGHIJKLMNOPQRSTUVWXYZ". If there are no characters inside the "[]", it is equivalent to a "%".
\	The "escape" character. The next character is treated as a literal character, not a wildcard character.

RIP Filters

RIP filters control the dynamic routing information content of the router's IPX routing tables, and the content of its RIP broadcasts, by filtering on the network number of each entry in an IPX RIP packet. RIP filter patterns may contain wildcards and combinations of service name and type.

A pattern is added to a RIP filter with the command:

```
add ipx rip=filter-number network=network action={include|
    exclude} [entry=entry-number]
```

The NETWORK parameter can be specified as a hexadecimal number or a numeric wildcard expression. An entry in a RIP filter can be modified with the command:

```
set ipx rip=filter-number entry=entry-number
    [network=network} [action={include|exclude}]
    [newentry=entry-number]
```

An entry can be deleted from a RIP filter with the command:

```
delete ipx rip=filter-number entry={entry-number|all}
```

The RIP filters and the patterns currently defined, and the number of matches, can be displayed with the command:

```
show ipx rip[=filter-number]
```

SAP Filters

SAP filters control the dynamic service information content of the router's IPX routing tables, and the content of its SAP broadcasts, by filtering on the service name and the service type of each entry in an IPX SAP packet. SAP filter patterns may contain wildcards and combinations of network numbers.

A pattern is added to a SAP filter with the command:

```
add ipx sap=filter-number service=service [type=service-type]
   action={include|exclude} [entry=entry-number]
```

The SERVICE parameter can be specified as an ASCII string or a string wildcard expression. The TYPE parameter can be specified as a known service type name, a hexadecimal number or a numeric wildcard expression. An entry in a SAP filter can be modified with the command:

```
set ipx sap=filter-number entry=entry-number
   [service=service] [type=service-type] [action={include|
   exclude}] [newentry=entry-number]
```

An entry can be deleted from a SAP filter with the command:

```
delete ipx sap=filter-number entry={entry-number|all}
```

The SAP filters and the patterns currently defined, and the number of matches, can be displayed with the command:

```
show ipx sap[=filter-number]
```

Global Versus Circuit-Specific Filters

Defining a filter does not automatically enable the filter. The filter must be assigned either globally or to a specific IPX circuit and traffic direction (transmit or receive). To assign a RIP or SAP filter globally use the command:

```
set ipx {grip|gsap}={filter-number|none}
```

To assign a RIP or SAP filter to a specific circuit and traffic direction, use either of the commands:

```
add ipx circuit=circuit... [inrip=filter-number|none]
   [outrip=filter-number|none] [insap=filter-number|none]
   [outsap=filter-number|none]
set ipx circuit=circuit... [inrip=filter-number|none]
   [outrip=filter-number|none] [insap=filter-number|none]
   [outsap=filter-number|none]
```

Dial-on-Demand IPX

The router's implementation of IPX supports dial-on-demand routing on circuits over ISDN interfaces. An IPX circuit can be created to use a PPP interface configured for dial-on-demand. When IPX packets are available for transmission, PPP initiates an ISDN call to the remote router. When the call is connected, the IPX circuit is activated and the data packets are transmitted over the circuit. When the circuit has been inactive for a set period of time, the IPX circuit is deactivated and PPP disconnects the ISDN call.

By default, RIP and SAP messages are sent over IPX circuits configured for dial-on-demand using the standard defined in RFC 1582. Another method for configuring on-demand circuits is to statically define the routes and services, using the commands:

```
add ipx route
add ipx service
```

Static or RFC 1582 demand learned routes and services are not affected by the normal ageing mechanisms which delete entries in the routing and service tables that have not been updated for a set period of time. When a circuit is disabled, the hop count of any static route or service using the circuit is set to 16, which effectively disables the route or service. When the module is re-enabled, the hop count is restored to its original value.

When a circuit is configured for dial-on-demand operation and static routes and services are not being used, change broadcasts must be enabled for both RIP and SAP. To gain the maximum benefit from IPX dial-on-demand, we recommend that change broadcasts be enabled only in combination with RIP and SAP filters.

If change broadcasts are enabled on a circuit, the circuit is brought up (an ISDN call is made) each time there is a change in the RIP or SAP database in the routers at either end of the circuit. RIP and SAP filters can be used to minimise the number and size of broadcast packets by filtering out entries from the RIP and SAP change broadcasts that are not required by the router at the remote end of the circuit. The combination of change broadcasts with RIP and SAP filters is an important technique for effective dial-on-demand IPX.

If general broadcasts are enabled on a circuit, the circuit is brought up each time there is a general RIP or SAP broadcast (typically every 600 seconds). The regular broadcasts increase line costs due to the increased number of ISDN calls. In a stable network where changes to the number or addresses of stations and servers are rare, costs can be reduced by increasing the interval between general broadcasts so that broadcasts occur once or twice a day during off-peak hours. The disadvantage of this approach is that a station may attempt to connect to a service that is no longer present, resulting in an SPX connection request storm. For circuits configured for dial-on-demand operation, general broadcasts are not required at all as the protocol extensions defined in RFC 1582 make general broadcasts unnecessary.

When there has been no activity on the connection between a Novell file server and a workstation for a specified period of time (set in the file server's system configuration), the file server sends a series of *watchdog* packets to the workstation. If the workstation does not reply with a *keep-alive* packet the file server deletes the connection so that it can be used by another workstation. If the connection is deleted, the workstation must re-establish the connection and login to the file server in order to access services provided by the server. If the

connection between the file server and the workstation is a dial-on-demand link the router must bring up the link to allow the exchange of just a few packets for the watchdog/keep-alive dialogue, and incur the cost of making an ISDN call. To reduce this overhead, the router on the network to which the file server is attached can be configured to respond to the watchdog packets on behalf of the workstation. This process is called *spoofing*.

SPX Spoofing with Dial-on-Demand IPX

In addition to spoofing IPX watchdog packets, the IPX module also provides support for spoofing SPX watchdog packets.

When there is no traffic on the connection between an SPX client session and an SPX server, a series of SPX watchdog packets is sent by both the client and server to make sure the connection is still alive. If watchdog packets are not received in response to the watchdog packets transmitted, the client or server at the other end of the connection is assumed to be dead or unreachable and the SPX connection is terminated. These watchdog packets do not contain data, and since they are sent approximately every 30 seconds depending on the cost (time from one end to the other) of the route, they keep an on-demand ISDN call up and incur unnecessary cost. Since some database applications use this watchdogging mechanism to keep track of the validity of record locks, it is important that the spoofing of SPX watchdog packets is configured correctly on the routers. The default settings for SPX spoofing on the router are correct for most situations.

Two of the most familiar SPX applications are RCONSOLE and RPRINTER, which are included in Novell Netware. RPRINTER works with the default settings for SPX spoofing because there is no data exchange while the printer is not being sent a job. When RCONSOLE is displaying data (e.g. a constantly changing system load graph), it is transmitting data. However, when the ':' prompt is displayed, RCONSOLE is performing SPX watchdogging and the link drops.

The SPX spoofing works by monitoring the SPX traffic over the on-demand ISDN link and checking for client/server conversations that are watchdogging. If a watchdog packet from the local network is transmitted over the link, the router listens for the corresponding watchdog packet from the remote end of the link. This information is logged in the SPX spoofing table and if no subsequent data is detected in the SPX conversation, the ISDN call is disconnected. The router spoofs a remote SPX watchdog packet if it detects a local watchdog packet while the ISDN call is disconnected. If data is detected, the ISDN call is reconnected, the data is sent and the SPX spoof table is cleared.

If an SPX connection is terminated abnormally while the ISDN call is disconnected, the router detects the absence of the SPX watchdogs and optionally bring up the link so that the remote end learns about the change. The SPX table is cleared when the ISDN call is re-established, and is rebuilt to reflect the current SPX watchdogging state. If any data is detected for a watchdogging SPX connection, it is deleted from the SPX spoof table.

Optionally, SPX spoofing can terminate after a certain period, and the ISDN call can be reactivated at the end of this period. For example, RPRINTER behaves unpredictably if the ISDN call is not reactivated at the end of SPX spoofing.

Troubleshooting SPX Spoofing

SPX spoofing is a subject that requires the operator to fully understand the SPX protocol and how the router spoofs the SPX watchdog packets. This understanding is vital for getting the best results from the router.

SPX Connection Request Storm

The symptoms are:

1. An SPX client program on a PC cannot connect to a service even though the PC thinks it is there.
2. Output of the **show ipx spxspoof** command on the router at the PC end of the WAN link contains a warning message about an "SPX connection request storm" (Figure 36-1 on page 36-14).

Figure 36-1: Example output from the **show ipx spxspoof** command during an SPX connection request storm

```
IPX SPX spoof table
Local Network:Station:Skt:Conn   Remote Network:Station:Skt:Conn  State
-------------------------------------------------------------------------------
000056e7:08005aa1dc95:4028:6151  c0e72301:000000000001:8060:ffff  Candidate
  Circ:2    Local count:1        Remote count:0        Age:95s
  Info. - Connection request packet observed above.
000056e7:08005aa1dc95:4029:23b5  c0e72301:000000000001:8060:ffff  Candidate
  Circ:2    Local count:27       Remote count:0        Age:1s
  WARNING - SPX connection request packet storm occurring on above connection!
            SPX connection request packets are being thrown away.
            Can be caused by an incorrect static service socket number.
-------------------------------------------------------------------------------
```

Follow these steps to determine the appropriate solution:

1. Check that any configured static services have the correct socket number. Incorrect socket numbers in a static service make the PC think that the service is at a different socket (or place) than it really is. The connection request packets are thrown away by the server, and the PC continuously retries. The router transmits the first two SPX connection requests it sees for a single source network:station:socket:connection-ID, so the on-demand link is not held up continuously.

   Check and correct the static service entries in the routers on each side of the WAN link. See the **add ipx route**, **add ipx service**, **delete ipx route**, and **delete ipx service** commands.

2. Filters may be preventing either the PC's packets from reaching the server or the server from replying to the PC. This can happen if a traffic filter is defined in an intermediate router preventing communication between the client and the server in one direction or the other, or even both directions. At this intermediate router, the service has not been filtered out of SAP broadcasts. Either correct the traffic filter in the intermediate router, or filter the SAP information out at the intermediate router.

3. The information in the router service table has somehow been corrupted in transit over the WAN link. Execute the **reset ipx** command on page 36-50 at both ends of the WAN link to refresh the IPX routing and service tables. Make sure the correct route and service information has been propagated to the router at each end of the WAN link by using the **show ipx route** command on page 36-72 and the **show ipx service** command on page 36-75.

If propagation is not reliable, check your router configuration for the relevant protocol and physical link (i.e. PPP, X.25, and ISDN). If the problem is definitely a line quality issue, talk to your telecommunications provider.

If these procedures do not resolve the problem, contact your authorised distributor or reseller.

PPP Idle Timer or X.25 SVC Inactivity Timer Set Too Low

The symptoms are:

1. ISDN or X.25 SVC calls are made repeatedly, in combination with:
2. Output of the **show ipx calllog** command on page 36-60 shows that SPX watchdog packets are causing frequent ISDN calls.

Follow these steps to determine the appropriate solution:

1. The inactivity/idle timeout on the physical link layer is too low. This causes the physical link to drop before the SPX spoof table has a complete record of all the SPX connections across the WAN link. As the router does not know about the SPX connection that is watchdogging, it activates the link for SPX watchdog packets.

   As the SPX watchdog packet is based on the cost (transit time) of the route between the PC and the server, increase the PPP idle or X.25 inactivity timeout to at least three times the cost of the route from the PC to the server. This cost can be found by obtaining the network number of the service from its address. The address is shown in the output from the **show ipx service** command on page 36-75. The cost can then be looked up from the relevant entry from the output of the **show ipx route** command on page 36-72. The router itself defaults to setup values for these timeouts and routes that are adequate for most situations.

If the above procedures do not resolve the problem, contact your authorised distributor or reseller.

Extended PING for IPX

The extended **ping** command on page 22-135 of Chapter 22, Internet Protocol (IP) supports Novell IPX and can be used test the connectivity between two IPX network stations to determine whether each station can "see" the other one. *Echo Request* packets are sent to the destination address and responses are recorded. The command:

```
ping [[ipxaddress=]network:station] [delay=seconds]
    [length=number] [number={number|continuous]
    [pattern=hexnum] [sipxaddress=network:station]
    [screenoutput={yes|no}] [timeout=number]
```

initiates the transmission of PING packets. Any parameters not specified use the defaults configured with a previous invocation of the command:

```
set ping [[ipxaddress=]network:station] [delay=seconds]
    [length=number] [number={number|continuous]
    [pattern=hexnum] [sipxaddress=network:station]
    [screenoutput={yes|no}] [timeout=number]
```

As each response packet is received a message is displayed on the terminal and the details are recorded. The default configuration and summary information can be displayed with the command:

```
show ping
```

The command:

```
stop ping
```

is used to halt a PING that is in progress. For a detailed description of the extended PING command, see the **ping** command on page 22-135 of Chapter 22, Internet Protocol (IP).

Counters and the MIB

The IPX module maintains a Management Information Base (MIB) containing counters and system configuration or operational parameters which are required to manage the IPX module.

The IPX MIB should not be confused with the IP MIB, which can be read using SNMP. At present the IPX MIB within the router cannot be read using SNMP.

The MIB counters can be accessed using:

```
show ipx counter[=option]
```

Selected parts of the MIB can be displayed by specifying one of the options CIRCUIT, GATEWAY or ROUTES. For example, the command:

```
show ipx count=circuit
```

displays counters relating to IPX circuits.

Configuration Examples

Examples in this section show the following configurations:

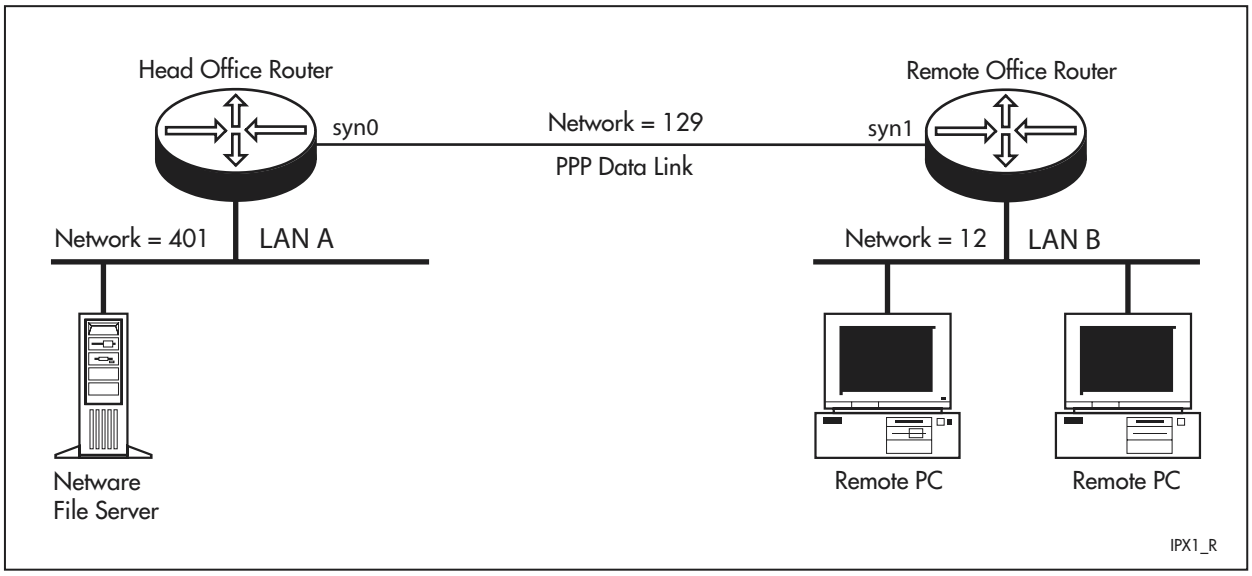
- **Basic IPX Setup**
- **IPX Dial-On-Demand**

Basic IPX Setup

This example shows the way in which a LAN is connected to other sites over a synchronous wide area link based on the Point-to-Point Protocol (PPP). A common application is to route packets (e.g. Email) between a remote site and a central site (Figure 36-2 on page 36-17).

Some interface and port types mentioned in this example may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the AR400 Series Router Hardware Reference.

Figure 36-2: Example configuration for a basic Novell IPX network.



The Head Office LAN is assigned network number 401, the Remote Office LAN is assigned network number 12, and the PPP link is assigned network number 129. These numbers must be unique across the particular internet. The PPP link is connected to synchronous interface 0 on the Head Office router and synchronous interface 1 on the Remote Office router.

Table 36-4: Example configuration parameters for an IPX network

Configuration Parameter	Head Office Router	Remote Office Router
Ethernet interface	eth0	eth0
Ethernet encapsulation	802.3	802.3
Novell network number for Ethernet	401	12
IPX circuit over Ethernet	1	1
PPP interface	ppp0	ppp0
Novell network number for PPP	129	129
IPX circuit over PPP	2	2

To function correctly, a Novell internetwork requires that each individual network be given a network number which is unique across this internetwork. This includes all LANs as well as each PPP link. All routers and fileservers on the same network must use the same number. Check to see what numbers your fileservers are using. Many schemes exist to ensure that numbers are kept unique. Using the full hexadecimal representation of the IP address is one method. Using the telephone numbers of each location is another.

To configure a basic wide area Novell internet

1. Configure the PPP link.

See "Configuring a PPP link" on page 15-34 of Chapter 15, Point-to-Point Protocol (PPP) for a step-by-step example of how to set up a PPP link. If management of either router using Telnet is required, then TCP/IP must be configured first. See "Basic IP Setup over PPP" on page 22-54 of Chapter 22, Internet Protocol (IP).

On the Head Office router, create a PPP interface to use synchronous interface 0, using the command:

```
create ppp=0 over=syn0
```

On the Remote Office router, create a PPP interface to use synchronous interface 1, using the command:

```
create ppp=0 over=syn1
```

2. Initialise the IPX routing module.

Enable the IPX routing module and purge the IPX static database to clear any old configuration information no longer required, using the following commands on each router:

```
enable ipx
```

3. Define IPX circuits for each interface to be used for IPX routing.

On the Head Office router define one circuit for the Ethernet interface and one circuit for the PPP interface, using:

```
add ipx circ=1 int=eth0 netw=401 encap=802.3
add ipx circ=2 int=ppp0 netw=129
```

On the Remote Office router define one circuit for the Ethernet interface and one circuit for the PPP interface, using:

```
add ipx circ=1 int=eth0 netw=12 encap=802.3
add ipx circ=2 int=ppp0 netw=129
```

The routers are now configured and capable of exchanging routes and service information.

The procedure for adding an additional interface of the same type is identical to that described above for ppp0. The **add ppp acservice** command on page 15-54 of Chapter 15, Point-to-Point Protocol (PPP) is used to create a new PPP interface and link the PPP module to the physical interface required.

To add a new interface type, such as Frame Relay, create an interface of the required type and add an IPX circuit to use the interface. For example, to create Frame Relay interface 1 to use synchronous port 1 on the Head Office router, use the command:

```
create fr=1 over=syn1
```

In the case of Frame Relay, a separate circuit must be added for each DLCI over which IPX is to run. To assign IPX circuit 3 to Frame Relay DLC 2 with a network number of 56, use the command:

```
add ipx circuit=3 interface=fr1 dlci=2 network=56
```

See "Configuration Examples" on page 14-19 of Chapter 14, Frame Relay for more detail examples of configuring Frame Relay.

4. **Add filters (optional).**

The basic setup described so far can be modified in several ways. It may be desirable to filter out certain networks or stations. As an example, the IPX module could be configured to exclude a particular local station (on the Ethernet interface) as well as an entire remote network (on the PPP interface). To exclude a local station (MAC address 12-34-56-78-9A-BC), use:

```
add ipx exclusion=42:123456789abc
```

To filter out another remote network (number 21) use:

```
add ipx exclusion=21
```

To disable the IPX module in the Remote Office router from advertising any local print servers, but to accept all remote services use:

```
add ipx sap=1 serv=* type=printserver act=exclude
add ipx sap=1 serv=* type=* act=include
set ipx circ=2 insap=1
```

5. **Test the configuration.**

Examine the route and service tables on each router. The route table contains paths from each Novell device that advertises routes (for example file servers) and other routers. The service table lists all services (for example file services, print services, etc.) that devices are advertising. To display the route table on the Head Office router use:

```
show ipx route
```

This command produces a display similar to Figure 36-3 on page 36-20. To display the service table, use:

```
show ipx service
```

This command produces a display similar to Figure 36-4 on page 36-20. ACCOUNTS is Netware 386 file server on LAN A and TYPISTS is a Netware 286 file server on LAN B.

Test that a workstation on one LAN can access a file server on the other LAN by attaching to the remote file server.

Figure 36-3: Example output from the **show ipx routes** command for a basic Novell IPX network

```
IPX routes
Network   Nexthop                  Circuit      Hops   Cost      Uptime  Type
-------------------------------------------------------------------------------
00000401  Local                      1 (eth0)     1       1       85973  Local
00000129  Local                      2 (ppp0)     1       1       85973  Local
00000012  00000129:0000cd000d26      2 (ppp0)     2       1       85973  RIP
-------------------------------------------------------------------------------
```

Figure 36-4: Example output from the **show ipx services** command for a basic Novell IPX network

```
IPX services
Name                                                                    Age
  Address                       Server type            Circuit      Hops  Defined
-------------------------------------------------------------------------------
ACCOUNTS                                                                0
  00007500:000000000001:0451    0004:FileServer          1 (eth0)     1   SAP
ACCOUNTS                                                                0
  00007500:000000000001:8104    0107:RConsole            1 (eth0)     1   SAP

TYPISTS                                                                 0
  00000012:0080488018d8:0451    0004:FileServer          2 (ppp0)     3   SAP
-------------------------------------------------------------------------------
```

The actual contents of the table varies with the number and type of file servers present in the network, but the general principle is that there should be a route from each router to the other, and all services shown as local (i.e. via eth0) on one router, should also be visible on the other router, but over the PPP link.

6. Save the configuration.

Save the new dynamic configuration as a script called ipx.cfg and set the router to use this script on restart.

```
set config=ipx.cfg
```


```
set config=ipx.cfg
```


Troubleshooting Basic Connections

No routes are visible to the remote router

1. Check the PPP link is active; see "Configuring a PPP link" on page 15-34 of Chapter 15, Point-to-Point Protocol (PPP) for a troubleshooting guide, or type:

   ```
   show ppp
   ```

   The display should look like Figure 36-5 on page 36-21.

Figure 36-5: Example output from the **show ppp** command for a basic Novell IPX network

```
Name            Enabled  ifIndex  Over                  CP           State
-------------------------------------------------------------------------
ppp0              YES    4                              IPXCP        OPENED
                                  syn0                  LCP          OPENED
-------------------------------------------------------------------------
```

The state of the IPX control protocol (IPXCP) should be "OPENED". If not, then the fault lies with the connection between the two routers, or the PPP configuration at either end.

2. Check that the IPX circuits are correctly configured on each router by typing:

   ```
   show ipx circuit
   ```

   For the Head Office router, the display should look like Figure 36-6 on page 36-21. The important points are that there are two circuits, and that the network numbers are correct. If not, then repeat steps 1 through 3.

Figure 36-6: Example output from the **show ipx circuit** command for a basic Novell IPX network

```
IPX CIRCUIT information

Name ........................ Circuit 1
Status ...................... enabled
Interface ................... eth0   (802.3)

Network number .............. 00000401
Station number .............. 0000cd000d34
Link state .................. up
Cost in Novell ticks ........ 1
Type20 packets allowed ...... no
On demand ................... no

Spoofing information
Keep alive spoofing ......... no
SPX watch dog spoofing ...... no
On SPX connection failure .... UPLINK
On end of SPX spoofing ...... UPLINK

RIP broadcast information
Change broadcasts ........... yes
General broadcasts .......... yes
General broadcast interval ... 60 seconds
Maximum age ................. 180 seconds

SAP broadcast information
Change broadcasts ........... yes
General broadcasts .......... yes
General broadcast interval ... 60 seconds
Maximum age ................. 180 seconds

Filter information
Filters ..................... none
```

3. Contact your authorised distributor or reseller for assistance.

Local workstations cannot access remote servers

This problem can be caused by a number of different events. The following give some of the most common:

1. Check that when the workstation is moved to the same LAN as the file server, that it is able to access the server. If not, the fault lies with the way the workstation or server is configured. Check with your network administrator.
2. Is the workstation using a packet driver to access its internal network card, or is it using another method such as WSGEN generated drivers? Ask your network administrator or contact your authorised distributor or reseller for assistance.

   Either method works with the router, but each has different configuration requirements on the workstation.
3. Care must be taken with the workstation NET.CFG file. Always specify the encapsulation (frame) as different LAN card drivers use different default encapsulations.
4. Does the remote file server appear in the IPX service table of the local router? If the server does not appear in the table, its presence cannot be advertised to the local LAN. Check this by typing:

   ```
   show ipx service
   ```

   This command produces a display similar to Figure 36-7 on page 36-22.

Figure 36-7: Example output from the **show ipx services** command for a basic Novell IPX network

```
IPX services
Name                                                                    Age
  Address                      Server type            Circuit      Hops  Defined
------------------------------------------------------------------------------
ACCOUNTS                                                                0
  00007500:000000000001:0451   0004:FileServer          1 (eth0)     1   SAP
ACCOUNTS                                                                0
  00007500:000000000001:8104   0107:RConsole            1 (eth0)     1   SAP

TYPISTS                                                                 0
  00000012:0080488018d8:0451   0004:FileServer          2 (ppp0)     3   SAP
------------------------------------------------------------------------------
```

Multiple servers may be present on either the local or remote LANs. The important thing is that the server in question must be in this table on the local router and there must be a route to the remote network. If there is, and it still does not work, contact your authorised distributor or reseller for assistance.

Check the route tables on the local and remote routers, using the command:

```
show ipx route
```

for the presence of networks on the remote side of the wide area network. If either of the remote networks is missing from the route tables, use the command:

```
reset ipx
```

which resets the IPX module and forces the routers to broadcast their routing and service tables.

IPX Dial-On-Demand

This example shows how to set up the router to provide a wide area internet based on Novell's IPX routing protocol with dial-on-demand access. IPX dial-on-demand uses ISDN to establish a connection between two LANs when there is IPX data to be transferred from one LAN to the other. When there is no more data to transfer, the ISDN call is disconnected.

Some interface and port types mentioned in this example may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the *AR400 Series Router Hardware Reference*.

For a pure dial-on-demand link, either static routes and services must be defined or the circuit **demand** parameter must be set to **on**, to minimise the normal repetitive exchange of route and service information between routers. When the circuit **demand** parameter is set to **on**, and there is a change of route or service information the link is activated to allow the exchange of RIP and SAP updates. If *general broadcasts* are enabled, the link is activated at regular intervals to allow the normal, regular exchange of route and service information. If the circuit **demand** parameter is set to **on**, general broad-casts should not be enabled. When both change and general broadcasts are enabled, the link is activated most of the time and is a permanent link.

This example illustrates the use of static routes and services, and change broadcasts. Figure 36-8 on page 36-23 shows a typical scenario, in which a PC at a remote site periodically accesses the Novell file server at a central site to read Email, transfer files or print documents on a high quality laser printer. Table 36-5 on page 36-24 lists parameter values that are used in the example.

Figure 36-8: Example configuration for IPX dial-on-demand

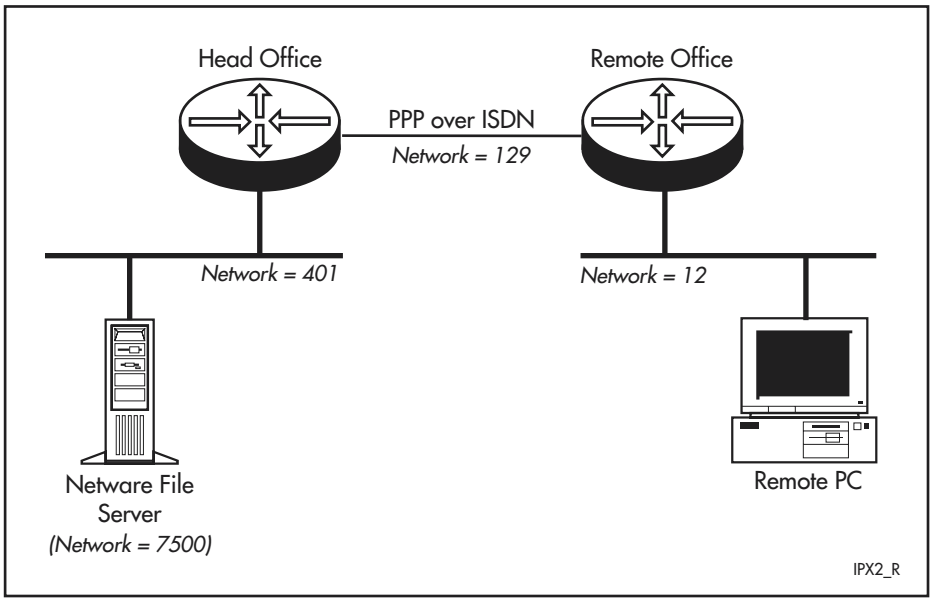


Table 36-5: Example configuration parameters for IPX dial-on-demand

Parameter	Head Office	Remote Office
Router name	HO	RO
Router MAC address	0000cd000d34	0000cd000d26
Router Ethernet interface	eth0	eth0
Novell Ethernet encapsulation	802.3	802.3
Novell network number for Ethernet	401	12
Router PPP interface	ppp0	ppp0
Novell network number for PPP	129	129
ISDN "phone" number	793281	367994
ISDN call name	RemoteC	RemoteC
Novell file server name	Accounts	-
File server internal network number	7500	-
File server internal station number	00000001	-
NCP socket address	0451	-

This example assumes the file server is running Netware 386 (version 3.1x or 4.0x). If your file server is running Netware 286 (version 2.15 or 2.2), there are no internal network number or internal station number. Instead, when adding the static service, use the network number of the cable segment (in this case 401) and the MAC address of the Ethernet card in the file server that connects the cable segment.

To find the file server's internal network number on Netware 386 use the 'config' command at the server ':' prompt. The first two lines returned give the server name and internal network number. The server internal station number used with Netware 386 is always 1. The socket address for all file servers is NCP (0451).

To function correctly, a Novell internetwork requires that each individual network be given a network number which is unique across this internetwork. This includes all LANs as well as each PPP link. All routers and file servers on the same network must use the same number. Check to see what numbers your file servers are using. Many schemes exist to ensure that numbers are kept unique. Using the full hexadecimal representation of the IP address or the telephone numbers of each location are two common schemes.

To configure a dial-on-demand IPX network

1. Create the ISDN calls.

To use dial-on-demand, an ISDN call must be defined on each router so that either router may initiate a call to transfer data. Set the ISDN call profile appropriate for the ISDN service provider. For example, to use the Australian profile on Basic Rate interface BRI 0, on router HO use the command:

```
set q931=bri0 profile=aus num1=793281
```

On router RO, use the command:

```
set q931=bri0 profile=aus num1=367994
```

On the Head Office router, create a call to the Remote Office router:

```
add isdn call=remotec number=367994 prec=in outsub=local
   searchsub=local
```

On the Remote Office router, create a call to the Head Office router:

```
add isdn call=remotec number=793281 prec=out outsub=local
   searchsub=local
```

Confirm that the calls have been defined correctly, using the command:

```
show isdn call
```

on each router.

2. **Create PPP interfaces to use the ISDN calls.**

PPP is used on the ISDN call just defined. PPP provides the link layer protocol for IPX. Configure the PPP interface for demand calls. On the Head Office router, create PPP interface 0 to use the ISDN call:

```
create ppp=0 over=isdn-remotec idle=on
```

On the Remote Office router, create PPP interface 0 to use the ISDN call:

```
create ppp=0 over=isdn-remotec idle=on
```

Confirm that PPP has been configured correctly, using the commands:

```
show ppp
```

```
show ppp conf
```

on each router. The output from the **show ppp** command on page 15-98 of Chapter 15, Point-to-Point Protocol (PPP) should show PPP0, the correct ISDN call name and LCP in the "STARTING" state. The output from the **show ppp config** command on page 15-99 of Chapter 15, Point-to-Point Protocol (PPP) should show "Open on traffic" set to ON.

3. **Test the ISDN calls and PPP.**

On the Head Office router activate the ISDN call to the Remote Office:

```
activate isdn call=remotec
```

You should see activity on the D channel LED and on one of the B channel LEDs on the router. Check the ISDN call status with the command:

```
show isdn call
```

There should be an active call. Check the PPP status with the command:

```
show ppp
```

The PPP status should be **opened**. Remember that without traffic the ISDN call is disconnected after the timeout period which is 60 seconds by default. If there are no problems, proceed to step 4. If there are problems establishing the ISDN calls or the PPP link, do not proceed until the problems have been resolved.

4. **Initialise the IPX routing module.**

Enable the IPX routing module and purge the IPX static database to clear any old configuration information no longer required, using the following commands on each router:

```
enable ipx
```

5. **Create IPX circuits on the Head Office router.**

   On the Head Office router define two circuits, one for the Ethernet interface and one for the PPP interface:

   ```
   add ipx circ=1 int=eth0 net=401 encap=802.3
   add ipx circ=2 int=ppp0 net=129 demand=on
   ```

   By setting the **demand** parameter to **on**, RIP and SAP change broadcasts are automatically enabled. To explicitly set them, use the command:

   ```
   set ipx circ=2 ripchange=yes sapchange=yes
   ```

   The amount of change traffic can be further minimised on the Head Office router by adding RIP and SAP filters to reduce the number and size of broadcasts (which activate ISDN calls). Note that service names are case sensitive:

   ```
   add ipx rip=0 net=7500 act=include
   add ipx sap=0 serv=accounts type=file act=include
   set ipx circ=2 outrip=0 outsap=0
   ```

   Otherwise, on the Head Office router, add a static route to the Remote Office router:

   ```
   add ipx route=12 circ=2 nexth=129:0000cd000d26
   ```

   When entering addresses, it is not necessary to enter leading zeros. For example, “0000cd000d34” can be entered as “cd000d34”.

6. **Create IPX circuits on the Remote Office router.**

   On the Remote Office router define two circuits, one for the Ethernet interface and one for the PPP interface:

   ```
   add ipx circ=1 int=eth0 netw=12 encap=802.3
   add ipx circ=2 int=ppp0 netw=129 demand=on
   ```

   If static routes and services are to be used on the Remote Office router, add a static route to the Head Office router and a service to access the Head Office file server:

   ```
   add ipx route=7500 circ=2 nexth=129:cd000d34 hops=3
   add ipx service=accounts address=7500:1:0451 type=file
      circ=2 hops=3
   ```

7. **Test the configuration.**

   On the Head Office router use the command:

   ```
   show ipx circuit=2
   ```

   This command produces a display similar to Figure 36-9 on page 36-27. There should be a circuit to the remote router over network 129. The *Status* field should be “enabled”, the *On demand* field should be “yes” and the *Link state* should be “down”.

Figure 36-9: Example output from the **show ipx circuit** command for the Head Office router in a dial-on-demand IPX network

```
IPX CIRCUIT information

Name ........................ Circuit 2
Status ...................... enabled
Interface ................... ppp0
Network number .............. 00000129
Station number .............. 0000cd000d34
Link state .................. down
Cost in Novell ticks ......... 1
Type20 packets allowed ....... no
On demand ................... yes

Spoofing information
Keep alive spoofing ......... yes
Keep alive timer ............. 60 minutes
SPX watch dog spoofing ....... yes
SPX timer ................... infinity minutes
On SPX connection failure .... UPLINK
On end of SPX spoofing ....... UPLINK

RIP broadcast information
Change broadcasts ............ yes
General broadcasts ........... no

SAP broadcast information
Change broadcasts ............ yes
General broadcasts ........... no

Filter information
Inwards RIP filter ........... none
Outwards RIP filter .......... 0
Inwards SAP filter ........... none
Outwards SAP filter .......... 0
Inwards TRAFFIC filter ....... none
Outwards TRAFFIC filter ...... none
```

Check that the filters have been correctly defined, using the commands:

```
show ipx rip
show ipx sap
```

Check the routes using the command:

```
show ipx route
```

This command produces a display similar to Figure 36-10 on page 36-28. There should be a route to the remote network (network 12) to which the PC is connected, as well as routes to the local file server and the networks attached to the router (networks 401 and 129).

Figure 36-10: Example output from the **show ipx route** command for the Head Office router in a dial-on-demand IPX network

```
IPX routes
Network   Nexthop                 Circuit      Hops   Cost      Uptime  Type
-------------------------------------------------------------------------------
00000012  00000129:0000cd000d26     2 (ppp0)     2       1       85973  Static
00007500  00000401:0000c06b035a     1 (eth0)     2       3        8475  RIP
00000401  Local                     1 (eth0)     1       1       85973  Local
00000129  Local                     2 (ppp0)     1       1       85973  Local
-------------------------------------------------------------------------------
```

Check the services using the command:

```
show ipx service
```

This command produces a display similar to Figure 36-11 on page 36-28. The *Accounts* service should be listed as a file server.

Figure 36-11: Example output from the **show ipx service** command for the Head Office router in a dial-on-demand IPX network

```
IPX services
Name                                                                    Age
  Address                       Server type           Circuit      Hops  Defined
-------------------------------------------------------------------------------
ACCOUNTS                                                                0
  00007500:000000000001:0451    0004:FileServer         1 (eth0)     2   SAP
ACCOUNTS                                                                0
  00007500:000000000001:8104    0107:RConsole           1 (eth0)     3   SAP
-------------------------------------------------------------------------------
```

On the Remote Office router use the command:

```
show ipx circuit=2
```

This command produces a display similar to Figure 36-12 on page 36-29. There should be a circuit to the remote router over network 129. The *Status* field should be “enabled”, the *On demand* field should be “yes” and the *Link state* should be “down”.

Check the routes using the command:

```
show ipx route
```

This command produces a display similar to Figure 36-13 on page 36-29. There should be a route to the file server (network 7500) via the Head Office router (MAC address 0000cd000d34) on network 129. There are also routes to the networks attached to the router (networks 12 and 129). If RIP filters have been assigned on the Head Office router, then the only non-local route is a route to the file server.

Check the services using the command:

```
show ipx service
```

This command produce a display similar to Figure 36-14 on page 36-29. The *Accounts* service should be listed as a file server. Test that the PC on the Remote LAN can access the file server on the Head Office LAN by attaching to the file server.

Figure 36-12: Example output from the **show ipx circuit** command for the Remote Office router in a dial-on-demand IPX network

```
IPX CIRCUIT information

Name ........................ Circuit 2
Status ...................... enabled
Interface ................... ppp0
Network number .............. 00000129
Station number .............. 0000cd000d26
Link state .................. down
Cost in Novell ticks ........ 1
Type20 packets allowed ...... no
On demand ................... yes

Spoofing information
Keep alive spoofing ......... yes
Keep alive timer ............ 60 minutes
SPX watch dog spoofing ...... yes
SPX timer ................... infinity minutes
On SPX connection failure .... UPLINK
On end of SPX spoofing ...... UPLINK

RIP broadcast information
Change broadcasts ........... yes
General broadcasts .......... no

SAP broadcast information
Change broadcasts ........... yes
General broadcasts .......... no

Filter information
Filters ..................... none
```

Figure 36-13: Example output from the **show ipx route** command for the Remote Office router in a dial-on-demand IPX network

```
IPX routes
Network   Nexthop                  Circuit      Hops   Cost      Uptime  Type
--------------------------------------------------------------------------------
00000012  Local                      1 (eth0)     1       1       85973  Local
00007500  00000129:0000cd000d34      2 (ppp0)     2       1        8475  Static
00000129  Local                      2 (ppp0)     1       1       85973  Local
--------------------------------------------------------------------------------
```

Figure 36-14: Example output from the **show ipx service** command for the Remote Office router in a dial-on-demand IPX network

```
IPX services
Name                                                                  Age
  Address                       Server type            Circuit      Hops  Defined
--------------------------------------------------------------------------------
ACCOUNTS                                                              0
  00007500:000000000001:0451   0004:FileServer          2 (ppp0)     2   Static
--------------------------------------------------------------------------------
```

8. **Save the configuration.**

   Save the new dynamic configuration as a script called ipx-dod.cfg and set the router to use this script on restart.

   ```
   set config=ipx-dod.cfg
   ```

   ```
   set config=ipx-dod.cfg
   ```

Troubleshooting Dial-on-Demand

No routes are visible to the remote router

1. Check the PPP link is active; see "Configuring a PPP link" on page 15-34 of Chapter 15, Point-to-Point Protocol (PPP) for a troubleshooting guide, or type:

   ```
   show ppp
   ```

   The state of the IPX control protocol (IPXCP) should be "OPENED". If not, then the fault lies with the connection between the two routers, or the PPP configuration at either end of the link.

2. Check that the circuits are correctly configured on each router using the command:

   ```
   show ipx circuit
   ```

   For the Head Office router, the display should look like Figure 36-9 on page 36-27. The important points are that there are two circuits, and that the network numbers are correct. If not, then repeat steps 4, 5 and 6.

3. Check the filters on the Head Office router, using the commands:

   ```
   show ipx rip
   ```

   ```
   show ipx sap
   ```

   The filters should be correctly defined and the counters should be increasing.

4. Contact your authorised distributor or reseller for assistance.

Local workstations cannot access remote servers

This problem can be caused by a number of different events. The following give some of the most common:

1. Check that when the workstation is moved to the same LAN as the file server, that it is able to access the server. If not, the fault lies with the way the workstation or server is configured. Check with your network administrator.

2. Is the workstation using a packet driver to access its internal network card, or is it using another method such as WSGEN generated drivers? Ask your network administrator or contact your authorised distributor or reseller for assistance.

   Either method works with the router but each has different configuration requirements on the workstation.

3. Does the remote file server appear in the IPX service table of the local router? If the server does not appear in the table, its presence cannot be advertised to the local LAN. Check this by typing:

   ```
   show ipx service
   ```

   For the Remote Office router this should produce a display like Figure 36-14 on page 36-29. Multiple servers may be present on either the local or remote LANs. The important thing is that the server in question must be in this table on the local router and there must be a route to the remote network. If there is, and it still does not work, contact your authorised distributor or reseller for assistance.

Command Reference

This section describes the commands available on the router to enable, configure, control and monitor the IPX module.

Some interface and port types mentioned in this chapter may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the *AR400 Series Router Hardware Reference*.

The shortest valid command is denoted by capital letters in the Syntax section. See "Conventions" on page lxv of About this Software Reference in the front of this manual for details of the conventions used to describe command syntax. See Appendix A, Messages for a complete list of messages and their meanings.

add ipx circuit

Syntax

```
ADD IPX CIRCuit=circuit INTerface=interface
   NETwork=network [COSt=1..999] [DEmand={ON|OFF|YES|NO|
   True|False}] [DLCI=dlci] [ENCapsulation={802.2|802.3|
   EThii|SNap}] [INRip=filter-number|NONE]
   [OUTRip=filter-number|NONE] [INSap=filter-number|NONE]
   [OUTSap=filter-number|NONE] [Keepalive={ON|OFF|YES|NO|
   True|False|Endlessly|FOrever|Infinite|Infinitely|
   Indefinitely|NONstop|1..1440}]
   [Mioxcircuit=circuit-name] [RIPChange={ON|OFF|YES|NO|
   True|False}] [RIPTimer=0..99999] [SAPChange={ON|OFF|
   YES|NO|True|False}] [SAPTimer=0..99999] [SPXSpoof={ON|
   OFF|YES|NO|True|False|Endlessly|FOrever|Infinite|
   Infinitely|Indefinitely|NONstop|0..1440}]
   [SPXendspoof={Uplink|Donothing}] [SPXConfail={Uplink|
   Donothing}] [TYPE20={ON|OFF|YES|NO|True|False}]
```

where:

- *circuit* is a manager-assigned identifier for the circuit from 1 to 512.
- *interface* is the name of a valid interface.
- *network* is a valid Novell network number for the interface, expressed as a hexadecimal number. Leading zeros may be omitted.
- *dlci* is a Frame Relay Data Link Connection Identifier.
- *filter-number* is a number from 0 to 99.
- *circuit-name* is the name of a valid MIOX circuit from 1 to 15 characters long.

Description

This command adds an IPX circuit to the interface. Each circuit is associated with a unique network number. More than one circuit can be added to the same X.25, Frame Relay or Ethernet interface. An Ethernet interface (e.g. eth0, vlan1) may have up to four circuits, one for each encapsulation type. Each circuit must use a different encapsulation. X.25 and Frame Relay interfaces may have one circuit for each PVC or DLC available on the interface. Other interface types may have only one circuit. Valid interfaces are:

- eth (such as eth0)
- PPP (such as ppp0)
- VLAN (such as vlan1)
- FR (such as fr0, fr0-1)
- X.25 DTE (such as x25t0, x25t0-1)

The specified interface must already exist. To see a list of all currently available interfaces, use the **show interface** command on page 9-72 of Chapter 9, Interfaces.

The **network** parameter specifies a valid Novell network number. It does not accept a network number of 0, to prevent the advertisement of routes to network 0 and services available on network 0.

The **cost** parameter specifies the cost associated with the circuit. The default is 1 for Ethernet interfaces (e.g. eth0, vlan1) and 20 for other interface types.

The **demand** parameter does not enable dial-on-demand, but sets the values of other parameters to default values suitable for dial-on-demand or normal connections (Table 36-6). The values **on**, **yes** and **true** are equivalent. The values **off, no** and **false** are equivalent. The default is **off**. By default, RIP and SAP messages are sent over IPX circuits configured for dial-on-demand using the standard defined in RFC 1582

Table 36-6: IPX dial-on-demand default parameter settings

Parameter	DEMAND=ON	DEMAND=OFF
KEEPALIVE	ON (infinity)	OFF
RIPCHANGE	ON	ON
SAPCHANGE	ON	ON
RIPTIMER	0	60 seconds for Ethernet (e.g. eth0, vlan1)
RIPTIMER	0	600 seconds for interfaces other than Ethernet
SAPTIMER	0	60 seconds for Ethernet (e.g. eth0, vlan1)
SAPTIMER	0	600 seconds for interfaces other than Ethernet
RIP maximum age	0	180 seconds for Ethernet (e.g. eth0, vlan1)
RIP maximum age	0	1800 seconds for interfaces other than Ethernet
SAP maximum age	0	180 seconds for Ethernet (e.g. eth0, vlan1)
SAP maximum age	0	1800 seconds for interfaces other than Ethernet

The DLCI parameter applies to Frame Relay interfaces and specifies the Frame Relay DLC to use for the circuit.

The **encapsulation** parameter specifies the encapsulation to be used for the specified Ethernet interface — 802.2, 802.3, Ethernet II or SNAP. A number of circuits, each with a different network number and encapsulation, can be added to the same Ethernet (e.g. eth0, vlan1) interface. The default is 802.3. The **encapsulation** parameter is invalid for other interface types.

The **inrip** parameter specifies a RIP filter to be applied to RIP packets received on the circuit. The specified RIP filter must already exist. If **none** is specified, no RIP filtering is applied to RIP packets received on the circuit. The default is **none**.

The **outrip** parameter specifies a RIP filter to be applied to RIP packets transmitted over the circuit. The specified RIP filter must already exist. If **none** is specified, no RIP filtering is applied to RIP packets transmitted over the circuit. The default is **none**.

The **insap** parameter specifies a SAP filter to be applied to SAP packets received on the circuit. The specified SAP filter must already exist. If **none** is specified, no SAP filtering is applied to SAP packets received on the circuit. The default is **none**.

The **outsap** parameter specifies a SAP filter to be applied to SAP packets transmitted over the circuit. The specified SAP filter must already exist. If **none** is specified, no SAP filtering is applied to SAP packets transmitted over the circuit. The default is **none**.

The **keepalive** parameter determines whether IPX keep-alive spoofing is on or off for the interface and how long spoofing should continue for a given connection. Keep-alive spoofing should be enabled on an interface that is used for dial-on-demand IPX connections (e.g. an ISDN interface). Setting the **keepalive** parameter to **on** sets a default spoofing time of 60 minutes, otherwise the time may be specified in minutes, as a value from 1 to 1440 (which is 24 hours). The values **on, yes** and **true** are equivalent. Setting the **keepalive** parameter to **off** disables spoofing. The values **off, no** and **false** are equivalent. Setting the **keepalive** parameter to **infinitely** enables spoofing with no timeout. The values **endlessly, forever, infinite, indefinitely** and **nonstop** are equivalent. The default is **off** if the **demand** parameter is set to **off**, or **infinitely** if the **demand** parameter is set to **on**.

The **mioxcircuit** parameter specifies the name of the MIOX circuit over the X.25 DTE interface, that carries the IPX circuit. The MIOX circuit must already exist for the X.25 DTE interface. The **interface** parameter must specify the X.25 DTE interface over which the MIOX circuit is defined, in the form:

```
INTERFACE=X25Tn
```

where n is the number of the X.25 DTE logical interface from 0 to 7.

The **ripchange** and **sapchange** parameters specify whether to send RIP and SAP change broadcasts on the interface. The values **on, yes** and **true** are equivalent. The values **off, no** and **false** are equivalent. The default is **off** for both parameters.

The **type20**, DEMAND, **ripchange** and **sapchange** parameters affect an entire interface.

The **riptimer** and **saptimer** parameters define the time intervals at which RIP and SAP packets are periodically sent on the circuit. The value specified is the number of seconds between regular broadcasts, and should be the same for all stations on a given network. The default in both cases is 60 seconds for

Ethernet interfaces (e.g. eth0, vlan1) and 600 seconds for other interface types. A value of 0 disables the transmission of RIP or SAP update packets on the particular interface, but delta updates continue.

Setting the value of a RIP or SAP timer prevents general broadcasts from being sent, but does not stop change broadcasts from being sent.

The value of the RIP and SAP timers must be the same for all stations on a network. The default should **not** be changed under normal circumstances.

The **spxspoof, spxendspoof** and **spxconfail** parameters determine the behaviour of SPX spoofing over the circuit. SPX spoofing is valid when the circuit is an on-demand circuit.

The **spxspoof** parameter determines whether SPX watchdog spoofing is enabled or disabled for the circuit and how long spoofing should continue for the circuit.

Setting the **spxspoof** parameter to **on, yes** or **true** sets SPX spoofing to continue indefinitely. If a time is specified, SPX spoofing is enabled for the specified time, in minutes. Setting the **spxspoof** parameter to **off** disables SPX watchdog spoofing. The values **off, no** and **false** are equivalent. Setting the **spxspoof** parameter to **infinitely** enables SPX watchdog spoofing with no timeout. The values **endlessly, forever, infinite, indefinitely** and **nonstop** are equivalent. The default is **off** if the **demand** parameter is set to **off**, or **infinitely** if the **demand** parameter is set to **on**.

The **spxendspoof** parameter determines whether the on-demand link should be re-established when SPX spoofing stops. If **spxendspoof** is set to **uplink** the call is re-established. If **spxendspoof** is set to **donothing** the link is not re-established. The default is **uplink**.

The **spxconfail** parameter determines whether the on-demand link should be re-established when an SPX connection fails. This occurs if the router does not receive SPX watchdog packets for 3 times the interval at which watchdog packets are generated locally. If **spxconfail** is set to **uplink** the call is re-established. If **spxconfail** is set to **donothing** the link is not re-established. The default is **uplink**.

The PPP idle timeout must be set to approximately 3 times the route cost or SPX spoofing does not function, and the line stays up, or the call goes up and down continuously.

The **type20** parameter specifies whether to broadcast NETBIOS type 20 packets on the interface. The default is **off**.

Examples

To add network 12345678 on eth0 as circuit 1 with default settings for all other parameters, use the command:

```
add ipx circ=1 interface=ETH0 network=12345678
```

To add network 98765432 on the same interface as circuit 2, with Ethernet II encapsulation, use the command:

```
add ipx circ=2 interface=ETH0 network=98765432
   encapsulation=ethii
```

To add network 10000001 to PPP interface 0 as circuit 3 running over a dial-on-demand Basic Rate ISDN link, use the command:

```
add ipx circ=3 interface=ppp0 network=10000001 demand=on
   ripch=yes sapch=yes inrip=0 outrip=1 insap=0 outsap=1
```

Related Commands

delete ipx circuit
disable ipx circuit
enable ipx circuit
set ipx circuit
show ipx circuit

add ipx exclusion

Syntax

`ADD IPX EXclusion=`*`network`*`[:`*`station`*`]`

where:

- *network* is a valid Novell network number, expressed as a hexadecimal number. Leading zeros may be omitted.
- *station* is the MAC address of a station on the network, expressed as a hexadecimal number. Leading zeros may be omitted.

Description

This command adds an exclusion filter to the exclusion filter list for networks or stations on a network. Any data received from a station or network that is on the exclusion list is discarded by the router. A station address of 0xFFFFFFFFFFFF excludes all stations on a network and effectively excludes that particular network. If the station address is not specified, a station address of 0xFFFFFFFFFFFF is assumed.

The inclusion list is checked first. If the packet passes this test then the exclusion list is checked. The packet is routed when it passes both tests. This method lets an inclusion be set for an entire network but individual stations on that network can still be excluded.

All IPX traffic including RIP and SAP packets is excluded. In comparison, RIP and SAP filters affect only RIP and SAP broadcast packets.

Note that if packets from any directly connected routers are excluded as a result of filtering, RIP and SAP broadcasts are not received from those routers.

The route and service tables are not automatically updated by this command. The IPX module may need to be reset before the route and service tables accurately reflect the updated filter list.

Examples

To exclude station 00-08-00-16-7B-23 from network 12345678 and all stations on network 401, use the commands:

```
add ipx exclusion=12345678:000800167b23
add ipx exclusion=401
```

Related Commands

add ipx inclusion
delete ipx exclusion
show ipx exclusion

add ipx inclusion

Syntax `ADD IPX INclusion=`*`network`*`[:`*`station`*`]`

where:

- *network* is a valid Novell network number, expressed as a hexadecimal number. Leading zeros may be omitted.
- *station* is the MAC address of a station on the network, expressed as a hexadecimal number. Leading zeros may be omitted.

Description This command adds an inclusion filter to the inclusion filter list for networks or stations on a network. Any data received from a station or network that is not on the inclusion list is discarded by the router. A station address of 0xFFFFFFFFFFFF includes all stations on a network and effectively includes that particular network. If the station address is not specified, a station address of 0xFFFFFFFFFFFF is assumed. To add an inclusion for a file server, specify the file server's internal network number, not the number of the network to which the file server is attached.

If no inclusion entries are specified, then all stations on all networks on each defined interface are included. The inclusion list is checked first. If the packet passes this test then the exclusion list is checked. The packet is routed when it passes both tests. This method allows an inclusion to be set for an entire network but individual stations on that network can still be excluded.

Note that if packets from any directly connected routers are excluded as a result of filtering, RIP and SAP broadcasts are not received from those routers.

The route and service tables are not automatically updated by this command. The IPX module may need to be reset before the route and service tables accurately reflect the updated filter list.

Examples To include station 00-08-00-16-7B-23 from network 12345678 and all stations on network 401, use the commands:

```
add ipx inclusion=12345678:000800167b23
add ipx inclusion=401
```

Related Commands **add ipx exclusion**
delete ipx inclusion
show ipx inclusion

add ipx rip

Syntax

```
ADD IPX RIP=filter-number NETwork=network ACtion={Include|
   Exclude} [ENTry=entry-number]
```

where:

- *filter-number* is a number from 0 to 99.
- *network* is a valid Novell network number, expressed as a hexadecimal number. Leading zeros may be omitted. Wildcard characters are allowed.
- *entry-number* is the position of this entry in the filter.

Description

This command adds a pattern to an IPX RIP filter. The exact pattern should not already exist in the filter. If the filter does not exist, it is created.

The **rip** parameter specifies the number of the filter to which the pattern is to be added.

The **network** parameter specifies the network address for the pattern. It may contain a numeric wildcard expression. Valid characters are ":*%[]0123456789ABCDEF".

The **action** parameter specifies the action to take when the pattern matches an entry in a RIP packet. If **include** is specified, the entry is added to the routing table. If **exclude** is specified, the entry is not added to the routing table.

The **entry** parameter specifies the position in the filter where this pattern is to be inserted, pushing all other entries down. If the value specified is greater than the number of the last entry in the filter, the entry is added to the end of the filter. The default is to add the pattern to the end of the filter. The last entry is always an implicit *match all* pattern and an exclusion action.

Examples

To exclude routing information for all networks with network numbers beginning with "000CF3" received on any router interfaces, use the commands:

```
add ipx rip=1 network=000cf3* act=exclude
add ipx rip=1 network=* act=include
set ipx grip=1
```

To apply the same filter to just incoming packets on circuit 1, use the command:

```
set ipx circ=1 inrip=1
```

instead of **set ipx grip**.

Related Commands

delete ipx rip
set ipx circuit
set ipx rip
show ipx
show ipx rip

add ipx route

Syntax

```
ADD IPX ROUte=network CIRCuit=circuit
   NEXthop=network:station [Hops=1..15] [COST=1..999]
```

where:

- *network* is a valid Novell network number for the network being added as a static route, expressed as a hexadecimal number. Leading zeros may be omitted.
- *circuit* is a manager-assigned identifier for the circuit from 1 to 64.
- *network:station* is the address of the next router to send packets to for this route, expressed as a pair of hexadecimal numbers. Leading zeros may be omitted.

Description

This command adds a static route to the IPX route table. This route is not deleted by any ageing mechanism, but is re-broadcast along with other routes. If route (either RIP or static) already exists to the network, the old route is replaced by the new static route.

The **network** parameter specifies a valid Novell network number. It does not accept a network number of 0, to prevent the advertisement of routes to network 0 and services available on network 0.

The network specified by the **route** parameter must be unique for each circuit because route table lookups are carried out on network number only. The circuit specified by the **circuit** parameter must already exist. The network number specified for the **nexthop** parameter must be a network number for the circuit. The **hops** parameter is the number of hops assigned for this route. The default is 2. The **cost** parameter specifies the cost associated with this route. The default is determined by the circuit cost, set with the **cost** parameter in the **add ipx circuit** command. The default is the circuit cost + 1.

Examples

To add a route to network 2345 on circuit 3, use the command:

```
add ipx route=2345 circ=3 nexthop=123:0000cd000d26
```

Related Commands

delete ipx route
show ipx route

add ipx sap

Syntax

```
ADD IPX SAP=filter-number SERvice=service
   [TYPE=service-type] ACtion={Include|Exclude}
   [ENTry=entry-number]
```

where:

- *filter-number* is a number from 0 to 99.
- *service* is the name of the service 1 to 48 characters long with no embedded spaces.
- *service-type* is the name of a recognised IPX service (Table 36-8 on page 36-41) or an IPX service type, expressed as a hexadecimal number. Leading zeros may be omitted. Wildcard characters are allowed.
- *entry-number* is the position of this entry in the filter.

Description

This command adds a pattern to an IPX SAP filter. The exact pattern should not already exist in the filter. If the filter does not exist, it is created.

The **sap** parameter specifies the number of the filter to which the pattern is to be added.

The **service** parameter specifies the name of the service to be filtered. It may contain a string wildcard expression.

The **type** parameter specifies the type of the service to be filtered. The value may be a recognised IPX service (Table 36-8 on page 36-41), a service type expressed as a hexadecimal number, or a numeric wildcard. If **type** is specified, **service** must also be specified.

The **action** parameter specifies the action to take when the pattern matches an entry in a SAP packet. If **include** is specified, the entry is added to the service table. If **exclude** is specified, the entry is not added to the service table.

The **entry** parameter specifies the position in the filter where this pattern is to be inserted, pushing all other entries down. If the value specified is greater than the number of the last entry in the filter, the entry is added to the end of the filter. The default is to add the pattern to the end of the filter. The last entry is always an implicit *match all* pattern and an exclusion action.

Examples

To include all service advertisements received via IPX circuit 1 but to advertise only print services in SAP broadcasts on circuit 1, use the commands:

```
add ipx sap=1 service=* type=print action=include
add ipx sap=1 service=* type=* action=exclude
set ipx circuit=1 outsap=1
```

Related Commands

delete ipx sap
set ipx circuit
set ipx sap
show ipx
show ipx sap

add ipx service

Syntax

```
ADD IPX SERvice=service ADdress=network:station:socket
   TYPE=service-type CIRCuit=circuit [HOPS=1..15]
```

where:

- *service* is the name of the service 1 to 48 characters long with no embedded spaces. The first character must be alphabetic (A–Z).
- *network:station:socket* is the full address of the service provider, including a valid Novell network number, the MAC address of the station, and the socket number (Table 36-7). All values are expressed as hexadecimal numbers. Leading zeros may be omitted.
- *service-type* is the name of a recognised IPX service (Table 36-8 on page 36-41) or an IPX service type, expressed as a hexadecimal number. Leading zeros may be omitted.
- *circuit* is a manager-assigned identifier for the circuit from 1 to 64.

Description

This command adds an IPX static service to the service table which is not deleted by the SAP ageing mechanism or the number of hops in the RIP table. A valid route must exist to the service or the router does not advertise the existence of the service.

The **address** parameter specifies the full address of the service provider. It does not accept a network number of 0, to prevent the advertisement of routes to network 0 and services available on network 0. If the service provider is a Novell file server the network address is the file server's internal network number, not the network number of the network to which the server is attached, and the station address is always 00000001. Table 36-7 lists some common IPX services and their socket numbers.

Table 36-7: Common IPX sockets

Socket Name	Socket Number (hexadecimal)
Advertising Print Server	0x8060
Btrieve	0x8059
File Server	0x0451
Named Pipes	0x9100
NDS Replica	0x4006
NetExplorer	0x401f
Netware Connect	0x1b90
Netware LANalyser Agent	0x0000
Netware Management Agent 1.5	0x2f90
Netware SQL	0x805b
NMS Console	0x0000
Remote Console	0x8104
Remote NLM Spawn (RSPAWN)	0x9085
Time Synchronisation	0x4005

The **type** parameter specifies the type of service. The value may be the name of a recognised service (Table 36-8), or a service type expressed as a hexadecimal number.

Table 36-8: Common IPX services

Service Name	Service Type (hexadecimal)
AdvPrintServer	0x0047
ArchiveServer	0x0009
BTrieve	0x004b
FileServer	0x0004
HMIHubs	0x0239
JobServer	0x0005
NamedPipe	0x009a
NDSReplica	0x0278
NetExplorer	0x0237
NetSQL	0x004c
NetwareConnect	0x024e
NLanalyserAgent	0x023a
NMA-1-5	0x0233
NMSConsole	0x026a
PrintQueue	0x0003
PrintServer	0x0007
RBridgeServer	0x0024
RConsole (Netware 386)	0x0107
Rspawn	0x9000
TimeSync	0x026b
Unknown	0x0000
Wildcard	0xffff

The **hops** parameter is the number of hops to this service, and defaults to 2.

Examples

To add the Accounts Department file server on a remote network accessed via circuit 3 to the service table, use:

```
add ipx service=accounts address=123:00000001:0451
   type=fileserver
```

Related Commands

delete ipx service
add ipx sap
delete ipx sap
set ipx sap
show ipx sap
show ipx service

delete ipx circuit

Syntax `DELete IPX CIRCuit=`*`circuit`*

where *circuit* is a manager-assigned identifier for the circuit from 1 to 512

Description This command deletes a circuit used by the IPX module. The circuit must already exist as the result of an **add ipx circuit** command. Any static routes using this circuit are disabled.

Note that if static routes have been defined which use the circuit, deleting the circuit and reusing the circuit number may lead to unpredictable results. It is a good idea, therefore, to delete any static routes or services using the circuit before deleting the circuit itself.

Examples To delete circuit 2, use the command:

```
delete ipx circuit=2
```

Related Commands

add ipx circuit
disable ipx circuit
enable ipx circuit
set ipx circuit
show ipx circuit

delete ipx exclusion

Syntax

`DELete IPX EXclusion=`*`network`*`[:`*`station`*`]`

where:

- *network* is a valid Novell network number, expressed as a hexadecimal number. Leading zeros may be omitted.
- *station* is the MAC address of a station on the network, expressed as a hexadecimal number. Leading zeros may be omitted.

Description

This command deletes an exclusion filter from the exclusion filter list for networks or stations on a network. Any data received from a station or network that is on the exclusion list is discarded by the router. A station address of 0xFFFFFFFFFFFF excludes all stations on a network and effectively excludes that particular network. If the station address is not specified, a station address of 0xFFFFFFFFFFFF is assumed. The exclusion specified must match exactly an entry in the exclusion filter list. The exclusion filter list can be displayed with the **show ipx exclusion** command.

Note that if packets from any directly connected routers are excluded as a result of filtering, RIP and SAP broadcasts are not received from those routers.

The route and service tables are not automatically updated by this command. The IPX module may need to be reset before the route and service tables accurately reflect the updated filter list.

Examples

To delete the exclusion for station 00-08-00-16-7B-23 on network 12345678, use the command:

```
delete ipx exclusion=12345678:000800167b23
```

To delete the exclusion for all stations on network 12345678, use either of the following commands:

```
delete ipx exclusion=12345678
delete ipx exclusion=12345678:ffffffffffff
```

Related Commands

add ipx exclusion
add ipx inclusion
show ipx exclusion

delete ipx inclusion

Syntax `DELete IPX INclusion=`*`network`*`[:`*`station`*`]`

where:

- *network* is a valid Novell network number, expressed as a hexadecimal number. Leading zeros may be omitted.
- *station* is the MAC address of a station on the network, expressed as a hexadecimal number. Leading zeros may be omitted.

Description This command deletes an inclusion filter from the inclusion filter list for networks or stations on a network. Any data received from a station or network that is not on the inclusion list is discarded by the router. A station address of 0xFFFFFFFFFFFF includes all stations on a network and effectively includes that particular network. If the station address is not specified, a station address of 0xFFFFFFFFFFFF is assumed. The inclusion specified must match exactly an entry in the inclusion filter list. The inclusion filter list can be displayed with the **show ipx inclusion** command.

If no inclusion entries are specified, then all stations on all networks on each defined interface are included. The inclusion list is checked first. If the packet passes this test then the exclusion list is checked. The packet is routed when it passes both tests. This method lets an inclusion be set for an entire network but individual stations on that network can still be excluded.

Note that if packets from any directly connected routers are excluded as a result of filtering, RIP and SAP broadcasts are not received from those routers.

The route and service tables are not automatically updated by this command. The IPX module may need to be reset before the route and service tables accurately reflect the updated filter list.

Examples To delete the inclusion for station 00-08-00-16-7B-23 on network 12345678, use the command:

```
delete ipx inclusion=12345678:000800167b23
```

Related Commands

add ipx exclusion
add ipx inclusion
show ipx inclusion

delete ipx rip

Syntax `DELete IPX RIP=`*`filter-number`* `ENTRY={`*`entry-number`*`|ALL}`

where:

- *filter-number* is a number from 0 to 99.
- *entry-number* is the position of this entry in the filter.

Description This command deletes a pattern from an IPX RIP filter. The exact pattern must already exist in the filter.

The **rip** parameter specifies the number of the filter from 0 to 99 from which the pattern is to be deleted.

The **entry** parameter specifies the number of the pattern being deleted from the filter. If **all** is specified, all patterns are deleted from the filter.

Examples To delete pattern 3 from RIP filter 1, use the command:

```
delete ipx rip=1 entry=3
```

Related Commands
add ipx rip
set ipx rip
show ipx rip

delete ipx route

Syntax `DELete IPX ROUte=`*`network`*

where *network* is a valid Novell network number for the network being deleted as a static route, expressed as a hexadecimal number. Leading zeros may be omitted.

Description This command deletes a static route from the IPX route table. The route must already exist as an IPX static route. The network specified by the **route** parameter must match exactly a network in the IPX route table.

Examples To delete the route to network 2345, use:

```
delete ipx route=2345
```

Related Commands
add ipx route
show ipx route

delete ipx sap

Syntax `DELete IPX SAP=`*`filter-number`* `ENTry={`*`entry-number`*`|ALL}`

where:

- *filter-number* is a number from 0 to 99.
- *entry-number* is the position of this entry in the filter.

Description This command deletes a pattern from an IPX SAP filter. The exact pattern must already exist in the filter.

The **sap** parameter specifies the number of the filter from 0 to 99 from which the pattern is to be deleted.

The **entry** parameter specifies the number of the pattern being deleted from the filter. If **all** is specified, all patterns are deleted from the filter.

Examples To delete all patterns from SAP filter 1, use the command:

```
delete ipx sap=1 entry=all
```

Related Commands **add ipx sap**
set ipx sap
show ipx sap

delete ipx service

Syntax `DELete IPX SERvice=`*`service`* `TYPE=`*`service-type`*

where:

- *service* is the name of the service 1 to 48 characters long with no embedded spaces. The first character must be alphabetic (A–Z).
- *service-type* is the name of a recognised IPX service (Table 36-8 on page 36-41) or an IPX service type, expressed as a hexadecimal number. Leading zeros may be omitted.

Description This command deletes an IPX static service from the service table. The service must already exist. The **type** parameter specifies the type of service. The value may be the name of a recognised service (Table 36-8 on page 36-41), or a service type expressed as a hexadecimal number. The service name and type must exactly match a static service entry in the service table.

Examples To delete the Accounts service, use the command:

```
delete ipx service=accounts type=fileserver
```

Related Commands
add ipx service
add ipx sap
delete ipx sap
set ipx sap
show ipx sap
show ipx service

disable ipx

Syntax `DISable IPX`

Description This command disables the IPX module. All IPX packet reception, processing and forwarding process are disabled.

The hop count of any static routes or services is set to 16, which effectively disables the routes or services. When the module is re-enabled, the hop count is restored to its original value.

Related Commands
enable ipx
show ipx

disable ipx circuit

Syntax `DISable IPX CIRCuit=`*`circuit`*

where *circuit* is a manager-assigned identifier for the circuit from 1 to 512

Description This command disables a circuit used by the IPX module. Packets are no longer transmitted or received over the circuit, but the configuration information is retained.

When a circuit is disabled, the hop count of any static route or service using the circuit is set to 16, which effectively disables the route or service. When the circuit is re-enabled, the hop count is restored to its original value.

Examples To disable circuit 1, use the command:

```
disable ipx circ=1
```

Related Commands
add ipx circuit
delete ipx circuit
enable ipx circuit
set ipx circuit
show ipx circuit

enable ipx

Syntax `ENAble IPX`

Description This command enables the IPX routing module, allowing IPX packet reception, processing and forwarding.

The hop count of any static routes or services is set to 16, which effectively disables the routes or services. When the module is re-enabled, the hop count is restored to its original value.

Related Commands
disable ipx
show ipx

enable ipx circuit

Syntax `ENAble IPX CIRCuit=`*`circuit`*

where *circuit* is a manager-assigned identifier for the circuit from 1 to 512

Description This command enables a circuit to be used by the IPX module. The circuit must have been previously disabled with the **disable ipx circuit** command. Packets can now be transmitted or received over the interface.

When a circuit is disabled, the hop count of any static route or service using the circuit is set to 16, which effectively disables the route or service. When the circuit is re-enabled, the hop count is restored to its original value.

Examples To enable circuit 1, use the command:

```
enable ipx circ=1
```

Related Commands
add ipx circuit
delete ipx circuit
disable ipx circuit
set ipx circuit
show ipx circuit

purge ipx

Syntax `PURge IPX`

Description This command removes all IPX configuration information and restore all defaults. This effectively disables the IPX module since all IPX circuits are purged. This command is typically used when the IPX module is disabled or when an immediate cessation of IPX operation does not cause a problem. The command should be used when making major changes to the IPX configuration to ensure that the new configuration is not affected by 'forgotten' parameters from the original configuration.

All IPX configuration information will be lost. Use this command with extreme caution. This command disables the IPX module. The IPX module must be re-enabled with the **enable ipx** command on page 36-48 before any other configuration commands will have any effect.

A log message is sent to the Logging facility.

Related Commands
reset ipx
show log in Chapter 60, Logging Facility

reset ipx

Syntax `RESET IPX`

Description This command reinitialises the IPX routing module. It should be used whenever the IPX routing module is first configured or whenever interfaces are changed. It causes the IPX module to rebuild all internal tables, such as the service and routing tables. Only the static routes and services are preserved. As far as other IPX routes are concerned, it is equivalent to turning the router off and then on.

This may disrupt communications until the internal data structures are rebuilt. If a file transfer is in progress through the router at the time the command is executed, it fails. Stations and file servers lose connectivity until the routing and service tables are rebuilt.

This command does not affect configuration information which is stored in non-volatile memory and this is retained during power outages. A log message is sent to the Logging facility.

Related Commands **purge ipx**
show log in Chapter 60, Logging Facility

set ipx circuit

Syntax

```
SET IPX CIRCuit=circuit [INTerface=interface]
   [NETwork=network] [COSt=1..999] [DEmand={ON|OFF|YES|NO|
   True|False}] [DLCI=dlci] [ENCapsulation={802.2|802.3|
   ETHii|SNap}] [INRip=filter-number|NONE]
   [OUTRip=filter-number|NONE] [INSap=filter-number|NONE]
   [OUTSap=filter-number|NONE] [Keepalive={ON|OFF|YES|NO|
   True|False|Endlessly|FOrever|Infinite|Infinitely|
   Indefinitely|NONstop|1..1440}]
   [Mioxcircuit=circuit-name] [RIPChange={ON|OFF|YES|NO|
   True|False}] [RIPTimer=0..99999] [SAPChange={ON|OFF|
   YES|NO|True|False}] [SAPTimer=0..99999] [SPXSpoof={ON|
   OFF|YES|NO|True|False|EndlesslY|FOrever|Infinite|
   Infinitely|Indefinitely|NONstop|0..1440}]
   [SPXEndspoof={Uplink|Donothing}] [SPXConfail={Uplink|
   Donothing}] [TYPE20={ON|OFF|YES|NO|True|False}]
```

where:

- *circuit* is a manager-assigned identifier for the circuit from 1 to 512.
- *interface* is the name of a valid interface.
- *network* is a valid Novell network number for the interface, expressed as a hexadecimal number. Leading zeros may be omitted.
- *dlci* is a Frame Relay Data Link Connection Identifier.
- *filter-number* is a number from 0 to 99.
- *circuit-name* is the name of a valid MIOX circuit from 1 to 15 characters long.

Description

This command changes the settings for an IPX circuit. Each circuit is associated with a unique network number. More than one circuit can be added to the same X.25, Frame Relay or Ethernet interface. An Ethernet interface (e.g. eth0, vlan1) may have up to four circuits, one for each encapsulation type. Each Ethernet circuit must use a different encapsulation. X.25 and Frame Relay interfaces may have one circuit for each PVC or DLC available on the interface. Other interface types may have only one circuit. Valid interfaces are:

- eth (such as eth0)
- PPP (such as ppp0)
- VLAN (such as vlan1)
- FR (such as fr0, fr0-1)
- X.25 DTE (such as x25t0, x25t0-1)

To modify a circuit's interface, the new interface type must be the same as that originally configured.

The **interface** parameter specifies a valid interface already assigned and configured. To see a list of current valid interfaces, use the **show interface** command on page 9-72 of Chapter 9, Interfaces.

The **network** parameter specifies a valid Novell network number. It does not accept a network number of 0, to prevent the advertisement of routes to network 0 and services available on network 0.

The **cost** parameter specifies the cost associated with the circuit. The default is 1 for Ethernet interfaces (e.g. eth0, vlan1) and 20 for other interface types.

The **demand** parameter does not enable dial-on-demand, but sets the values of other parameters to default values suitable for dial-on-demand or normal connections (Table 36-8 on page 36-41). The values **on**, **yes** and **true** are equivalent. The values **off, no** and **false** are equivalent. The default is **off**.

The **dlci** parameter applies to Frame Relay interfaces and specifies the Frame Relay DLC to use for the circuit.

The **encapsulation** parameter is required for Ethernet interfaces, and specifies the Ethernet encapsulation to be used for the specified network — 802.2, 802.3, Ethernet II or SNAP. A number of circuits, each with a different network number and encapsulation, can be added to the same Ethernet interface (e.g. eth0, vlan1). The default is 802.3. The **encapsulation** parameter is invalid for other interface types.

The **inrip** parameter specifies the new RIP filter to be applied to RIP packets received on the circuit. The specified RIP filter must already exist. If **none** is specified, no RIP filtering is applied to RIP packets received on the circuit. The default is **none**.

The **outrip** parameter specifies the new RIP filter to be applied to RIP packets transmitted over the circuit. the specified rip filter must already exist. if none is specified, no RIP filtering is applied to RIP packets transmitted over the circuit. The default is **none**.

The **insap** parameter specifies the new SAP filter to be applied to SAP packets received on the circuit. The specified SAP filter must already exist. If **none** is specified, no SAP filtering is applied to SAP packets received on the circuit. The default is **none**.

The **outsap** parameter specifies the new SAP filter to be applied to SAP packets transmitted over the circuit. The specified SAP filter must already exist. If **none** is specified, no SAP filtering is applied to SAP packets transmitted over the circuit. The default is **none**.

The **keepalive** parameter determines whether IPX keep-alive spoofing is on or off for the interface and how long spoofing should continue for a given connection. Keep-alive spoofing should be enabled on an interface that is used for dial-on-demand IPX connections (e.g. an ISDN interface). Setting the **keepalive** parameter to **on** sets a default spoofing time of 60 minutes, otherwise the time may be specified in minutes, as a value from 1 to 1440 (which is 24 hours). The values **on, yes** and **true** are equivalent. Setting the **keepalive** parameter to **off** disables spoofing. The values **off, no** and **false** are equivalent. Setting the **keepalive** parameter to **infinitely** enables spoofing with no timeout. The values **endlessly, forever, infinite, indefinitely** and **nonstop** are equivalent. The default is **off** if the **demand** parameter is set to **off**, or **infinitely** if the **demand** parameter is set to **on**.

The **mioxcircuit** parameter specifies the name of the MIOX circuit over the X.25 DTE interface, that carries the IPX circuit. The MIOX circuit must already exist for the X.25 DTE interface. The **interface** parameter must specify the X.25 DTE interface over which the MIOX circuit is defined, in the form:

```
interface=x25tn
```

where n is the number of the X.25 DTE logical interface from 0 to 7.

The **ripchange** and **sapchange** parameters specify whether to send RIP and SAP change broadcasts on the interface. The values **on**, **yes** and **true** are equivalent. The values **off**, **no** and **false** are equivalent. The default is **off** for both parameters.

The **type20**, **demand**, **ripchange**, and **sapchange** parameters affect an entire interface.

The **riptimer** and **saptimer** parameters define the time intervals at which RIP and SAP packets are periodically sent on the circuit. The value specified is the number of seconds between regular broadcasts, and should be the same for all stations on a given network. The default in both cases is 60 seconds for Ethernet interfaces (e.g. eth0, vlan1) and 600 seconds for other interface types. A value of 0 disables the transmission of RIP or SAP update packets on the particular interface, but delta updates still continue.

Note that setting the value of a RIP or SAP timer prevents general broadcasts from being sent, but does not stop change broadcasts from being sent.

The value of the RIP and SAP timers must be the same for all stations on a network. The default should **not** be changed under normal circumstances.

The **spxspoof, spxendspoof** and **spxconfail** parameters determine the behaviour of SPX spoofing over the circuit. SPX spoofing is valid when the circuit is an on-demand circuit.

The **spxspoof** parameter determines whether SPX watchdog spoofing is enabled or disabled for the circuit and how long spoofing should continue for the circuit. SPX spoofing should be enabled for an interface that is used for dial-on-demand IPX connections (e.g. an ISDN interface). Setting the **spxspoof** parameter to **on**, **yes** or **true** sets SPX spoofing to continue indefinitely. If a time is specified, SPX spoofing is enabled for the specified time, in minutes. Setting the **spxspoof** parameter to **off** disables SPX watchdog spoofing. The values **off**, **no** and **false** are equivalent. Setting the **spxspoof** parameter to **infinitely** enables SPX watchdog spoofing with no timeout. The values **endlessly**, **forever, infinite, indefinitely** and **nonstop** are equivalent. The default is **off** if the **demand** parameter is set to **off**, or **infinitely** if the **demand** parameter is set to **on**.

The **spxendspoof** parameter determines whether the on-demand link should be re-established when SPX spoofing stops. If **spxendspoof** is set to **uplink** the call is re-established. If **spxendspoof** is set to **donothing** the link is not re-established. The default is **uplink**.

The **spxconfail** parameter determines whether the on-demand link should be re-established when an SPX connection fails. This occurs if the router does not receive SPX watchdog packets for 3 times the interval at which watchdog packets are generated locally. If **spxconfail** is set to **uplink** the call is re-established. If **spxconfail** is set to **donothing** the link is not re-established. The default is **uplink**.

The PPP idle timeout must be set to approximately 3 times the route cost or SPX spoofing does not function, and the line stays up, or the call goes up and down continuously.

The **type20** parameter specifies whether to broadcast NETBIOS type 20 packets on the interface. The default is **off**.

Examples To broadcast NETBIOS type 20 packets on circuit 1, use the command:

```
set ipx circ=1 type20=on
```

To disable dial-on-demand for circuit 3, use the command:

```
SET IPX CIRC=3 DEMAND=OFF
```

Related Commands
add ipx circuit
delete ipx circuit
disable ipx circuit
enable ipx circuit
show ipx circuit

set ipx grip|gsap

Syntax `SET IPX {GRip|GSap}={`*`filter-number`*`|NONE}`

where *filter-number* is a number from 0 to 99

Description This command attaches or detaches filters to the global RIP and SAP filter points.

If **grip** is specified with a filter number, the specified RIP filter is added to the global RIP filter point. The specified RIP filter must already exist. If **grip** is specified with a value of **none**, all RIP filters are removed from the global RIP filter point.

If **gsap** is specified with a filter number, the specified SAP filter is added to the global SAP filter point. The specified SAP filter must already exist. If **gsap** is specified with a value of **none**, all SAP filters are removed from the global SAP filter point.

Examples To attach RIP filter 1 as a global RIP filter, use the command:

```
set ipx grip=1
```

Related Commands
add ipx circuit
set ipx circuit

set ipx rip

Syntax

```
SET IPX RIP=filter-number ENTry=entry-number
   [NETwork=network} [ACtion={Include|Exclude}]
   [NEWentry=entry-number]
```

where:

- *filter-number* is a number from 0 to 99.
- *network* is a valid Novell network number, expressed as a hexadecimal number, or a numeric wildcard.
- *entry-number* is the position of this entry in the filter.

Description

This command modifies a pattern in an IPX RIP filter. The filter must already exist and the exact pattern must already exist in the filter.

The **rip** parameter specifies the number of the filter which contains the pattern to be modified.

The **entry** parameter specifies the pattern in the filter to be modified.

The **network** parameter specifies the new network address for the pattern, and may include a numeric wildcard expression. Valid characters are ":*%[]^0123456789ABCDEF".

The **action** parameter specifies the new action to take when the pattern matches an entry in a RIP packet. If **include** is specified, the entry is added to the routing table. If **exclude** is specified, the entry is not added to the routing table.

The **newentry** parameter specifies the new position in the filter for this pattern, pushing all other entries down. If the value specified is greater than the number of the last entry in the filter, the entry is added to the end of the filter. The default is to add the pattern to the end of the filter.

Examples

To move pattern 3 of RIP filter 2 to the end of the filter, use the command:

```
set ipx rip=2 entry=3 newentry=99
```

Related Commands

add ipx rip
delete ipx rip
show ipx rip

set ipx sap

Syntax

```
SET IPX SAP=filter-number ENTry=entry-number
   [SERvice=service] [TYPE=service-type] [ACtion={Include|
   Exclude}] [NEWentry=entry-number]
```

where:

- *filter-number* is a number from 0 to 99.
- *entry-number* is the position of this entry in the filter.
- *service* is the name of the service 1 to 48 characters long with no embedded spaces.
- *service-type* is the name of a recognised IPX service (Table 36-8 on page 36-41) or an IPX service type, expressed as a hexadecimal number. Leading zeros may be omitted.

Description

This command modifies a pattern in an IPX SAP filter. The filter must already exist and the exact pattern must already exist in the filter.

The **SAP** parameter specifies the number of the filter which contains the pattern to be modified.

The **entry** parameter specifies the pattern in the filter to be modified.

The **service** parameter specifies the new name of the service to be filtered. It may contain a string wildcard expression.

The **type** parameter specifies the new type of the service to be filtered. The value may be a recognised IPX service (Table 36-8 on page 36-41), a service type expressed as a hexadecimal number, or a numeric wildcard. If **type** is specified, **service** must also be specified.

The **action** parameter specifies the new action to take when the pattern matches an entry in a SAP packet. If **include** is specified, the entry is added to the service table. If **exclude** is specified, the entry is not added to the service table.

The **newentry** parameter specifies the new position in the filter for this pattern, pushing all other entries down. If the value specified is greater than the number of the last entry in the filter, the entry is added to the end of the filter. The default is to add the pattern to the end of the filter.

Examples

To move pattern 3 of SAP filter 2 to the end of the filter, use the command:

```
set ipx sap=2 entry=3 newentry=99
```

Related Commands

add ipx sap
delete ipx sap
show ipx sap

show ipx

Syntax `SHow IPX`

Description This command displays the current software revision and status (enabled or disabled) of the IPX module (Figure 36-15, Table 36-9).

Figure 36-15: Example output from the **show ipx** command

```
IPX general configuration

Module Status ................... Enabled
Module version .................. 2.4

Filter information:
  Global RIP filter ............. None
  Global SAP filter ............. 1
  Global inclusions ............. None
  Global exclusions ............. None

Circuit information:
  Circuits ...................... 1
  Filter attachments ............ None

Route information:
  Looped networks ............... None
  Route table entries:
    Dynamic ..................... 4
    Local ....................... 1
    Static ...................... 1
    Total ....................... 6

Service information:
  File servers .................. 1
  Service table entries:
    Dynamic ..................... None
    Static ...................... 1
    Total ....................... 1
```

Table 36-9: Parameters in output of the **show ipx** command

Parameter	Meaning
Module Status	Whether the IPX module is enabled.
Module version	The major and minor versions of the IPX software.
Filter information	Summary information about filters.
Global RIP filter	The currently attached global RIP filter, or “None”.
Global SAP filter	The currently attached global SAP filter, or “None”.
Global inclusions	The number of entries in the IPX inclusion traffic filter table, or “None”.
Global exclusions	The number of entries in the IPX exclusion traffic filter table, or “None”.
Circuit information	Summary information about IPX circuits.
Circuits	The number of IPX circuits, or “None”.

Table 36-9: Parameters in output of the **show ipx** command (Continued)

Parameter	Meaning
Filter attachments	The number of RIP or SAP filters attached to circuits, or "None".
Route information	Summary information about IPX routes.
Looped networks	The number of equal cost network loops that the router knows about, or "None".
Route table entries	Summary information about entries in the route table.
Dynamic	The number of route table entries learned from RIP packets, or "None".
Local	The number of route table entries automatically defined from active circuits, or "None".
Static	The number of router table entries from static routes, or "None".
Total	The total number of route table entries, or "None".
Service information	Summary information about IPX services.
File servers	The number of Novell file servers that this router knows about, or "None".
Service table entries	Summary information about entries in the service table.
Dynamic	The number of service table entries learned from SAP packets, or "None".
Static	The number of service table entries from static services, or "None".
Total	The total number of service table entries, or "None".

Related Commands

add ipx circuit
delete ipx circuit
disable ipx
enable ipx
set ipx circuit
show ipx circuit

show ipx cache

Syntax `SHow IPX CAChe`

Description This command displays the contents of the IPX route cache (Figure 36-16). The route cache speeds the forwarding of IPX packets by maintaining session information about recently forwarded packets. When a packet is about to be forwarded after making all the prerequisite filtering and routing decisions, an entry is added to the IPX route cache. Further IPX packets are checked against sessions stored in the cache. If a match is found no further processing is required and the packet is forwarded to the known destination route. The cache is automatically purged every five minutes and on events such as routing and filtering changes.

The display lists, for each session, the full IPX address (network:station:socket) of the two IPX devices and the number of hits on the cache for that session.

Figure 36-16: Example output from the **show ipx cache** command

```
IPX routing cache

  c0e72301:000000000001:0451  001234dd:0000c0c06957:4003  9098
  001234dd:0000c0c06957:4003  c0e72302:000000000001:0451  1111
```

Related Commands show ipx route

show ipx calllog

Syntax `SHow IPX CALLlog[=`*`circuit`*`]`

where *circuit* is an IPX circuit number from 1 to 512

Description This command displays IPX packets that may have caused the activation of an on-demand link, and is intended as a tool for debugging ISDN or X.25 SVC calls. The router maintains a log of the headers of the last 40 IPX packets that were sent to an on-demand circuit while the physical link was down. The first 150 bytes of each packet are recorded. If the packet is smaller than 150 bytes, then the whole packet is recorded. The output is very similar to that obtained from a network packet analyser (Figure 36-18 on page 36-61). SPX packets with a Connection Control of 80 and a packet length of 42 are SPX watchdog packets (Figure 36-17 on page 36-60). SPX packets with a Destination Connection ID of 65535 (0xFFFF) are SPX connection requests. If there are a lot of connection request packets from the same source address and source connection ID in the call log, an SPX connection request packet storm is occurring (Figure 36-19 on page 36-62).

For more detailed information about IPX packet formats, refer to the following Novell documents:

IPX Router Specification, Document Version 1.2, Part Number 107-000029-001.

Novell's Guide to NetWare LAN Analysis (Second Edition), L. A. Chappell and D. E. Hakes, 1994. Novell Press. ISBN 0-7821-1362-1.

Figure 36-17: Example output from the **show ipx calllog** command for a watchdog packet

```
IPX On Demand Calls
------------------------------------------------------------------------
Circuit = 2 Time of Event: 16:29:31 26-Jul-1995
Reason: Packet transmitted.

CheckSum:           65535
Packet Length:      42
Transport Control:  2
Packet Type:        SPX
Destination:        00008991:08005aa1dc95:4009
Source:             c0e72301:000000000001:811e
Connection Control: 80( SYS )
Datastream Type:    0x00 (Client)
Source Con ID:      4512
Destination Con ID: 13706
Sequence Number:    0
Acknowledge Number: 2
Allocation Number:  2
------------------------------------------------------------------------
```

Figure 36-18: Example output from the **show ipx calllog** command for RIP and SAP broadcast packets

```
IPX On Demand Calls
--------------------------------------------------------------------------
Circuit = 2 Time of Event: 16:27:42 26-Jul-1995
Reason: Packet transmitted.

CheckSum:           65535
Packet Length:      40
Transport Control:  0
Packet Type:        RIP
Destination:        000000f1:ffffffffffff:0453
Source:             000000f1:0000cd000f25:0453

00 01 ff ff ff ff 00 00 00 00

.  .  .  .  .  .  .  .  .  .
--------------------------------------------------------------------------
Circuit = 2 Time of Event: 16:27:42 26-Jul-1995
Reason: Packet transmitted.

CheckSum:           65535
Packet Length:      288
Transport Control:  0
Packet Type:        SAP
Destination:        000000f1:ffffffffffff:0452
Source:             000000f1:0000cd000f25:0452

00 02 00 a1 50 57 52 43 48 55 54 45 5f 41 44 4d 49 4e 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 c0 e7 23 01 00 00 00 00 00 01 80 d8 00 03 00 04 41 44 4d 49
4e 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 c0 e7 23 01

.  .  .  .  P  W  R  C  H  U  T  E  _  A  D  M  I  N  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  #  .  .  .  .  .  .  .  .  .  .  .  .  A  D  M  I  .
N  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #  .  .
--------------------------------------------------------------------------
```

Figure 36-19: Example output from the **show ipx calllog** command for a connection request storm

```
IPX On Demand Calls
-----------------------------------------------------------------------
Circuit = 2 Time of Event: 16:29:31 26-Jul-1995
Reason: Packet transmitted.

CheckSum:            65535
Packet Length:       42
Transport Control:   2
Packet Type:         SPX
Destination:         00008991:08005aa1dc95:4009
Source:              c0e72301:000000000001:811e
Connection Control:  C0( SYS ACK )
Datastream Type:     0x00 (Client)
Source Con ID:       4518
Destination Con ID:  65535
Sequence Number:     0
Acknowledge Number:  2
Allocation Number:   2
-----------------------------------------------------------------------
Circuit = 2 Time of Event: 16:29:31 26-Jul-1995
Reason: Packet transmitted.

CheckSum:            65535
Packet Length:       42
Transport Control:   2
Packet Type:         SPX
Destination:         00008991:08005aa1dc95:4009
Source:              c0e72301:000000000001:811e
Connection Control:  C0( SYS ACK )
Datastream Type:     0x00 (Client)
Source Con ID:       4518
Destination Con ID:  65535
Sequence Number:     0
Acknowledge Number:  2
Allocation Number:   2
-----------------------------------------------------------------------
Circuit = 2 Time of Event: 16:29:31 26-Jul-1995
Reason: Packet transmitted.

CheckSum:            65535
Packet Length:       42
Transport Control:   2
Packet Type:         SPX
Destination:         00008991:08005aa1dc95:4009
Source:              c0e72301:000000000001:811e
Connection Control:  C0( SYS ACK )
Datastream Type:     0x00 (Client)
Source Con ID:       4518
Destination Con ID:  65535
Sequence Number:     0
Acknowledge Number:  2
Allocation Number:   2
-----------------------------------------------------------------------
```

show ipx circuit

Syntax `SHow IPX CIRCuit[=`*`circuit`*`]`

Description This command displays the configuration of all IPX circuits which have been configured for use by the IPX module (Figure 36-20, Table 36-10 on page 36-64).

Figure 36-20: Example output from the **show ipx circuit** command

```
IPX CIRCUIT information

Name ........................ Circuit 1
Status ...................... enabled
Interface ................... eth0   (802.3)
Network number .............. c0e7230f
Station number .............. 0000cd000d26
Link state .................. up
Cost in Novell ticks ........ 1
Type20 packets allowed ...... no
On demand ................... no

Spoofing information
Keep alive spoofing ......... no
SPX watch dog spoofing ...... no
On SPX connection failure .... UPLINK
On end of SPX spoofing ...... UPLINK

RIP broadcast information
Change broadcasts ........... yes
General broadcasts .......... yes
General broadcast interval ... 60 seconds
Maximum age ................. 180 seconds

SAP broadcast information
Change broadcasts ........... yes
General broadcasts .......... yes
General broadcast interval ... 60 seconds
Maximum age ................. 180 seconds

Filter information
Filters ..................... none
```

Table 36-10: Parameters in output of the **show ipx circuit** command

Parameter	Meaning
Name	The circuit name.
Status	The status of the circuit; one of "enabled" or "disabled".
Interface	The interface the circuit is using. For Ethernet interfaces (e.g. eth0, vlan1), the encapsulation is also specified. For an X.25 interface, the MIOX circuit name is specified. For a Frame Relay DLCI, the channel number or DLCI number is specified.
Network number	The network address.
Station number	The station address.
Link state	The status of the circuit; one of "up" or "down".
Cost in Novell ticks	The cost associated with the circuit.
Type20 packets allowed	Whether IPX Type20 packets are broadcast on the circuit; one of "yes" or "no".
On demand	Whether the circuit is configured for dial-on-demand; one of "yes" or "no".
Keep alive spoofing	Whether keep-alive spoofing is enabled on the circuit; one of "Yes" or "No".
Keep alive timer	The value of the keep-alive timer, in seconds. This field is present when keep alive spoofing is enabled.
SPX watch dog spoofing	Whether the circuit is configured for SPX watchdog spoofing; one of "Yes" or "No".
SPX timer	The SPX spoofing timeout timer, in seconds, or "Inf" if spoofing is enabled without a timeout. This field is present when SPX watchdog spoofing is enabled.
On SPX connection failure	The action to take when an SPX connection fails; one of "UPLINK" (bring up the ISDN link) or "DONOTHING" (do not bring up the ISDN link).
On end of SPX spoofing	The action to take when spoofing terminates; one of "UPLINK" (bring up the ISDN link) or "DONOTHING" (do not bring up the ISDN link).
Change broadcasts	Whether RIP or SAP change broadcasts are enabled on the circuit; one of "Yes" or "No".
General broadcasts	Whether RIP or SAP general broadcasts are enabled on the circuit; one of "Yes" or "No".
General broadcast interval	The value, in seconds, of the RIP or SAP general broadcast timer.
Maximum age	The maximum age, in seconds, of RIP or SAP entries in the route or service tables (respectively).
Filters	The number of filters attached to the circuit.

Related Commands

add ipx circuit
delete ipx circuit
disable ipx circuit
enable ipx circuit

show ipx counter

Syntax `SHow IPX COUnter[={CIRCUIT|GATEWAY|ROUTES}]`

Description This command displays the values of the counters in the IPX MIB. If an option is not specified all the IPX counters are displayed. The counters can be selectively displayed by specifying one of the options **circuit**, **gateway** or **routes**.

The **circuit** option displays information about IPX packets sent and received on the router interfaces that have been configured for IPX (Figure 36-21, Table 36-11 on page 36-66). The **gateway** option displays summary information about all IPX packets sent and received by the router (Figure 36-22 on page 36-67, Table 36-12 on page 36-67). The **route** option displays the number of bytes sent and received on each route known to the router (Figure 36-23 on page 36-68, Table 36-13 on page 36-68).

Figure 36-21: Example output from the **show ipx counter=circuit** command

```
IPX circuit counters

Circuit:   1 (ppp0)
inReceives:                    14        outPackets :           16
inOctets  :                   512        outOctets  :         2320
inDiscards:                     0
ripInTrigRequests :             1        ripOutTrigRequests  :            2
ripInTrigResponses:             2        ripOutTrigResponses :            2
ripInTrigAcks     :             2        ripOutTrigAcks      :            2
sapInTrigRequests :             1        sapOutTrigRequests  :            2
sapInTrigResponses:             2        sapOutTrigResponses :            6
sapInTrigAcks     :             6        sapOutTrigAcks      :            2

Circuit:   2 (eth0)
inReceives:                    22        outPackets :            3
inOctets  :                  2416        outOctets  :          122
inDiscards:                     0
```

Table 36-11: Parameters in output of the **show ipx counter**=CIRCUIT command

Parameter	Meaning
Circuit	The circuit number.
inReceives	The number of IPX packets received on this circuit.
inOctets	The number of input bytes received on this circuit.
inDiscards	The number of input packets discarded by this circuit because no input buffer space was available.
outPackets	The number of IPX packets transmitted over this circuit.
outOctets	The number of bytes transmitted over this circuit.
ripInTrigRequests	The number of RIP triggered request packets received by this circuit.
ripInTrigResponses	The number of RIP triggered response packets received by this circuit.
ripInTrigAcks	The number of RIP triggered acknowledgement packets received by this circuit.
ripOutTrigRequests	The number of RIP triggered request packets transmitted over this circuit.
ripOutTrigResponses	The number of RIP triggered response packets transmitted over this circuit.
ripOutTrigAcks	The number of RIP triggered acknowledgement packets transmitted over this circuit.
sapInTrigRequests	The number of SAP triggered request packets received by this circuit.
sapInTrigResponses	The number of SAP triggered response packets received by this circuit.
sapInTrigAcks	The number of SAP triggered acknowledgement packets received by this circuit.
sapOutTrigRequests	The number of SAP triggered request packets transmitted over this circuit.
sapOutTrigResponses	The number of SAP triggered response packets transmitted over this circuit.
sapOutTrigAcks	The number of SAP triggered acknowledgement packets transmitted over this circuit.

Figure 36-22: Example output from the **show ipx counter=gateway** command

```
IPX general counters

inReceives:                    18        outRequests:            2
inDiscards:                     0        outPackets:             0
inHdrErrors:                    0        outNoRoutes:            0
inUnknownSockets:               0        outForwarded:           0
inBadChecksums:                 0
inDelivers:                    18
inTooManyHops:                  0
inFiltered:                     0

NETBIOS packets:                0
RIP bad packets:                0
SAP bad packets:                0
```

Table 36-12: Parameters in output of the **show ipx counter=gateway** command

Parameter	Meaning
inReceives	The number of IPX packets received by the router.
inDiscards	The number of IPX packets discarded by the router because it was unable to process them. They are discarded at the IPX level and hence the reason is related to IPX in some way.
inHdrErrors	The number of packets received by the router with an invalid header.
inUnknownSockets	The number of packets received by the router for an unknown socket.
inBadChecksums	The number of packets received by the router with a bad checksum.
inDelivers	The number of packets delivered to a higher layer protocol.
inTooManyHops	The number of packets received by the router that had exceeded the hop count.
inFiltered	The number of packets received by the router that were discarded due to filtering.
outRequests	The number of IPX packets generated by the router.
outPackets	The number of IPX packets transmitted by the router, including packets generated by the router and packets forwarded by the router.
outNoRoutes	The number of IPX packets discarded because the router had no route to the final destination.
outForwarded	The number of IPX packets which the router has forwarded to either another router or to the final destination. These are generally NCP packets.
NETBIOS packets	The number of NETBIOS packets received by the router.
RIP bad packets	The number of invalid RIP packets received by the router.
SAP bad packets	The number of invalid SAP packets received by the router.

Figure 36-23: Example output from the **show ipx counter=routes** command

```
IPX route counters

Network   Bytes received       Bytes sent
----------------------------------------
c0e72301            2103             1024
c0e72302               0                0
c0e7230d               0                0
c0e7230e               0                0
c0e7230f            4096             2048
----------------------------------------
```

Table 36-13: Parameters in output of the **show ipx counter=routes** command

Parameter	Meaning
Network	The remote network address.
Bytes Received	The number of octets received from the remote network.
Bytes sent	The number of octets sent to the remote network.

Related Commands

add ipx route
delete ipx route

show ipx exclusion

Syntax `SHow IPX EXclusion`

Description This command displays the exclusion filter list (Figure 36-24, Table 36-14). If networks or stations are entered into this list, traffic from those networks or stations is excluded from the routing process.

Figure 36-24: Example output from the **show ipx exclusion** command.

```
IPX Exclusions
Exclusion                    Matches
-----------------------------------
00000012:ffffffffffff   0000000004
-----------------------------------
```

Table 36-14: Parameters in output of the **show ipx exclusion** command

Parameter	Meaning
Exclusion	The network and station address to be excluded.
Matches	The number of packets discarded by the filter.

Related Commands

add ipx exclusion
add ipx inclusion
delete ipx exclusion
delete ipx inclusion
add ipx sap
delete ipx sap
set ipx sap
show ipx inclusion
show ipx sap

show ipx inclusion

Syntax `SHow IPX INclusion`

Description This command displays the inclusion filter list (Figure 36-25, Table 36-15). If networks or stations are entered into this list, traffic from those networks or stations is included in the routing process.

Figure 36-25: Example output from the **show ipx inclusion** command

```
IPX Inclusions
Inclusion                    Matches
---------------------------------
c0e72301:ffffffffffff   0000002056
c0e7230f:ffffffffffff   0000001456
---------------------------------
```

Table 36-15: Parameters in output of the **show ipx inclusion** command

Parameter	Meaning
Inclusion	Network and station address to be included.
Matches	Number of packets forwarded by the filter.

Related Commands

add ipx exclusion
add ipx inclusion
delete ipx exclusion
delete ipx inclusion
add ipx sap
delete ipx sap
set ipx sap
show ipx exclusion
show ipx sap

show ipx rip

Syntax `SHow IPX RIP[=`*`filter-number`*`]`

where *filter-number* is a number from 0 to 99

Description This command displays information about IPX RIP filters. If a filter is specified, the patterns in the RIP filter are displayed. If a filter is not specified, the patterns in all RIP filters are displayed (Figure 36-26, Table 36-16).

Figure 36-26: Example output from the **show ipx rip** command

```
IPX RIP Filter

No AttachCount
       Ent Network                                    Action    Matches
 0-  1-------------------------------------------------------------------
         0 c0e7230[12]                               include         335
      virt *                                         exclude         731

       Requests: 1111         Passes: 380           Fails: 731
 1-NA--------------------------------------------------------------------
         0 B0905                                     include           0
      virt *                                         exclude           0

       Requests: 0            Passes: 0             Fails: 0
-------------------------------------------------------------------------
```

Table 36-16: Parameters in output of the **show ipx rip** command

Parameter	Meaning
No	The number of the filter.
AttachCount	The number of times the filter has been attached, globally or to a circuit.
Ent	The entry number in this filter for the pattern, or "virt" (catch-all filter added automatically).
Network	The network number or network wildcard for this pattern.
Action	The filter action; one of "include" or "exclude".
Matches	The number of RIP packet entries that have matched this pattern.
Requests	The number of RIP packet entries checked against the filter.
Passes	The number of RIP packet entries included by the filter.
Fails	The number of RIP packet entries excluded by the filter.

Related Commands

add ipx rip
delete ipx rip
set ipx rip

show ipx route

Syntax `SHow IPX ROUte`

Description This command displays all static and dynamic routes in the IPX route table (Figure 36-27, Table 36-17). Static entries are added to the route table using the **add ipx route** command. Dynamic entries are added by the RIP protocol. Since this relies on IPX for transport, an entry is added to the route table when it has not been subject to filtering. Static routes over circuits that have been disabled appear with the *Hops* field set to 16.

Figure 36-27: Example output from the **show ipx route** command

```
IPX routes
Network   Nexthop                 Circuit      Hops   Cost      Uptime  Type
------------------------------------------------------------------------------
c0e72301  c0e7230f:0000c0142745     1 (eth0)     2       1         200  Static
c0e72302  c0e7230f:0000c0204f54     1 (eth0)     2       3         200  RIP
c0e7230d  c0e7230f:0000c0142745     1 (eth0)     2       3         200  RIP
c0e7230e  c0e7230f:0000c0142745     1 (eth0)     2       3         200  RIP
c0e7230f  Local                     1 (eth0)     1       1         200  Local
------------------------------------------------------------------------------
```

Table 36-17: Parameters in output of the **show ipx route** command

Parameter	Meaning
Network	The remote network number.
Nexthop	The network number and station address of the next router on the path to the remote network.
Circuit	The circuit that the next hop is on.
Hops	The number of hops to the remote network.
Cost	The estimated time in 1/18th second, that a packet takes to reach the destination.
Uptime	The time in seconds that this route has been available.
Type	The type of route; one of "Static", "RIP" or "Local".

Related Commands
add ipx route
delete ipx route

show ipx sap

Syntax `SHow IPX SAP[=`*`filter-number`*`]`

where *filter-number* is a number from 0 to 99

Description This command displays information about IPX SAP filters. If a filter is specified, the patterns in the SAP filter are displayed. If a filter is not specified, the patterns in all SAP filters are displayed (Figure 36-28, Table 36-18).

Figure 36-28: Example output from the **show ipx sap** command

```
IPX SAP Filter

No AttachCount
   Ent Service
          Type                                          Action     Matches
 0-  1-------------------------------------------------------------------
     0 *ASLAN_HPJET*
          *                                             exclude        156
     1 CAVORTA
          *                                             exclude          0
     2 *ASLAN*
          *                                             include        625
     3 BOGUS
          *                                             exclude          0
  virt *
          *                                             exclude        308

       Requests: 1089         Passes: 625           Fails: 464
60-NA-------------------------------------------------------------------
     0 ALIAS
          FileServer,RConsole                         include          0
  virt *
          *                                             exclude          0

       Requests: 0            Passes: 0             Fails: 0
-------------------------------------------------------------------------
```

Table 36-18: Parameters in output of the **show ipx sap** command

Parameter	Meaning
No	The number of the filter.
AttachCount	The number of times the filter has been attached, globally or to a circuit.
Ent	The entry number in this filter for the pattern, or "virt" (catch-all filter added automatically).
Service	The name of the service, or a string wildcard expression, for this pattern.
Type	The type of the service, or a numeric wildcard, for this pattern.
Action	The filter action; one of "include" or "exclude".
Matches	The number of SAP packet entries that have matched this pattern.
Requests	The number of SAP packet entries checked against the filter.

Table 36-18: Parameters in output of the **show ipx sap** command (Continued)

Parameter	Meaning
Passes	The number of SAP packet entries included by the filter.
Fails	The number of SAP packet entries excluded by the filter.

Related Commands

add ipx exclusion
add ipx inclusion
add ipx sap
delete ipx exclusion
delete ipx inclusion
delete ipx sap
set ipx sap
show ipx exclusion
show ipx inclusion

show ipx service

Syntax `SHow IPX SERvice`

Description This command displays all static and dynamic services in the IPX service table (Figure 36-29, Table 36-19). Static entries are added to the service table using the **add ipx service** command. Dynamic entries are added by the SAP protocol. Since this relies on IPX for transport, an entry is added to the service table when it has not been subject to filtering. Static services over circuits that have been disabled appear with the *Hops* field set to 16.

Figure 36-29: Example output from the **show ipx service** command

```
IPX services
Name                                                                    Age
  Address                       Server type           Circuit     Hops  Defined
------------------------------------------------------------------------------
PWRCHUTE_ADMIN                                                          0
  c0e72301:000000000001:80d8    00a1:unknown            1 (eth0)     3  SAP
ENGINEERING                                                             0
  c0e72302:000000000001:0451    O004:FileServer         1 (eth0)     2  SAP
ENGINEERING                                                             0
  c0e72302:000000000001:8104    0107:RConsole           1 (eth0)     3  SAP
ADMIN                                                                   0
  c0e72301:000000000001:0451    0004:FileServer         1 (eth0)     2  SAP
ADMIN                                                                   0
  c0e72301:000000000001:8104    0107:RConsole           1 (eth0)     3  SAP
ADMIN_PS                                                                0
  c0e72301:000000000001:8060    0047:AdvPrintServer     1 (eth0)     3  SAP
BOGUS                                                                   0
  00000056:000000000001:0451    0004:FileServer       512 (eth0)    14  Static
0800090D16E703C2ADMIN_HPJET                                             0
  c0e7230f:0800090d16e7:400c    030c:unknown            1 (eth0)     2  SAP
------------------------------------------------------------------------------
```

Table 36-19: Parameters in output of the **show ipx service** command

Parameter	Meaning
Name	The name of the service.
Age	The age of this entry.
Address	The full network, station, socket address of the server.
Server Type	The hexadecimal value and name of the service being filtered.
Circuit	The circuit from which the service information was received.
Hops	The number of hops to the service.
Defined	The mechanism used to define the service; one of "SAP" or "Static".

Related Commands

add ipx service
delete ipx service

show ipx spxspoof

Syntax `SHow IPX SPXSpoof`

Description This command displays the current state of the SPX spoofing table which keeps track of the watchdogging SPX connections (Figure 36-30, Table 36-20).

Figure 36-30: Example output from the **show ipx spxspoof** command

```
IPX SPX spoof table
Local Network:Station:Skt:Conn   Remote Network:Station:Skt:Conn  State
--------------------------------------------------------------------------------
000056e7:08005aa1dc95:4026:a76c  c0e72301:000000000001:8060:ffff  Candidate
  Circ:2    Local count:1        Remote count:0        Age:122s
  Info. - Connection request packet observed above.
000056e7:08005aa1dc95:4027:a86c  c0e72301:000000000001:811e:ffff  Candidate
  Circ:2    Local count:1        Remote count:0        Age:122s
  Info. - Connection request packet observed above.
000056e7:08005aa1dc95:4028:6151  c0e72301:000000000001:8060:ffff  Candidate
  Circ:2    Local count:1        Remote count:0        Age:95s
  Info. - Connection request packet observed above.
000056e7:08005aa1dc95:4029:23b5  c0e72301:000000000001:8060:ffff  Candidate
  Circ:2    Local count:27       Remote count:0        Age:1s
  WARNING - SPX connection request packet storm occuring on above connection!
            SPX connection request packets are being thrown away.
            Can be caused by an incorrect static service socket number.
000056e7:08005aa1dc95:402a:24b5  c0e72301:000000000001:811e:ffff  Candidate
  Circ:2    Local count:26       Remote count:0        Age:1s
  WARNING - SPX connection request packet storm occuring on above connection!
            SPX connection request packets are being thrown away.
            Can be caused by an incorrect static service socket number.
000056e7:08005aa1dc95:402a:24b5  c0e72301:000000000001:811e:15c3  Valid
  Circ:2    Local count:2        Remote count:2        Age:2s     Valid:8s
000056e7:08005aa1dc95:4025:1fe0  c0e72301:000000000001:8060:ffff  Candidate
  Circ:2    Local count:1        Remote count:0        Age:125s
  Info. - Connection request packet observed above.
--------------------------------------------------------------------------------
```

Table 36-20: Parameters in output of the **show ipx spxspoof** command

Parameter	Meaning
Local Network:Station:Skt:Conn	The network number, station number, socket and connection ID for the local SPX station.
Remote Network:Station:Skt:Conn	The network number, station number, socket and connection ID for the Remote SPX station.
State	The state of the entry; one of "Candidate" or "Valid".
Circ	The circuit for which the SPX connection record is valid.
Local count	The number of local SPX watchdog packets seen for this SPX connection.
Remote count	The number of remote SPX watchdog packets seen for this SPX connection.
Age	The time, in centiseconds, since a local watchdog packet for this SPX connection was last seen.
Valid for	The length of time the entry has been valid.

The SPX packets originating locally are examined for SPX watchdog packets. When a watchdog packet is found, a record is added to the spoof table containing the SPX connection information from that packet. Traffic off the WAN is monitored for the complementary SPX watchdog packet from the remote end. During this listening phase, the record is in the 'candidate' state. When a complementary watchdog packet off the WAN is found and its information recorded, the record is said to be in the 'valid' state and PPP is told to time out the link for this traffic. Data packets seen on the SPX connection delete the record.

SPX connection requests also show up in the spoof table with a remote connection ID of 0xffff, which is a 'broadcast' value for the socket. They remain in the table for twenty minutes before being timed out, unless more broadcast packets are received. A storm of broadcast packets occurs if there is an incorrect socket in a static service. This is shown by a warning message under the appropriate entry.

Sometimes SPX candidates are created with only one SPX acknowledge observed. These entries are also timed out.

Related Commands

add ipx circuit
show ipx circuit