Chapter 1
Routing Novell IPX

1

This chapter describes Cisco's implementation of the Novell IPX routing protocol. You will find these topics and tasks described in this chapter:

- Overview of addressing in Novell IPX networks
- Basic Novell IPX routing configuration steps
- Processes for configuring optional parameters
- Steps for developing and using Novell IPX access lists
- Examples of controlling Novell broadcast messages and using various filters

Cisco's Implementation of Novell IPX

Novell IPX is a variation on Xerox Network Systems (XNS). One major difference between IPX and XNS is that they do not use the same Ethernet encapsulation format. A second difference is that IPX uses Novell's proprietary Service Advertisement Protocol (SAP) to advertise special network services. A file server is one instance of a service typically advertised.

Cisco's implementation of Novell's IPX protocol provides all of the functionality of a Novell "External Bridge" (Novell refers to their router functionality as bridging). As a Novell External Bridge, a Cisco router connects Ethernets and Token Rings, either directly or through high-speed serial lines (56 kbps to T1 speeds) or X.25. Novell workstations on any LAN, including those without a file server, or connects to Novell file servers on any other LAN. Novell sells an X.25 and a T1 interface capability. At this time, the Cisco X.25 and T1 support is not compatible with Novell. This means that Cisco routers must be used on both ends of T1 and X.25 circuits.

Novell Addresses

Novell node IDs are 48-bit quantities, represented by dotted triplets of four-digit hexadecimal numbers. A Novell router will have interfaces on more than one physical network (Ethernet, Token Ring, serial line, and so on). Physical networks are identified by 32-bit numbers, written in hexadecimal. These network numbers must be unique throughout a Novell internet. Since both the network number and the host address are needed to deliver

traffic to a host, addresses are usually given as network numbers, followed by host addresses, separated with dots. An example would be:

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

Here, the network number is *4a*, and the host address is *0000.0c00.23fe*.

Note: All numbers in the address, including the network number *4a*, are expressed in hexadecimal numbers.

Configuring Novell Routing

There are only two commands required to enable Novell IPX routing:

Step 1: Enable routing using the global configuration command **novell routing**.

Step 2: Assign Novell routing to a specific interface using the interface subcommand **novell network**.

All other configuration commands provide additional functionality or refinements. Each task is described in the following section. These descriptions are followed by applicable EXEC commands for monitoring and debugging Novell networks. Summaries of global configuration commands and interface subcommands described here appear at the end of this chapter.

Novell Configuration Restrictions

An interface takes as its Novell host address the hardware MAC address currently assigned to the interface. If later the MAC address is changed to some other value, the Novell node address automatically changes to the new address. Of course, connectivity will be lost for a while because of this change.

An optional address argument to the **novell routing** configuration command (see below) establishes the default Novell node address. This address is used as the Novell node address for any non-LAN interface, such as serial links.

Enabling Novell Routing

To enable or disable Novell routing, use the **novell routing** global configuration command. The full command syntax of this command follows.

novell routing [*host-address*]

no novell routing

The argument *host-address* is optional. If you do not specify an address, the MAC address of

the first Ethernet, Token Ring, or FDDI interface is used. If there are no satisfactory interfaces present, you must specify the host address argument using the optional *host-address* argument. The address must not be multicast. The **novell routing** command enables Novell RIP routing and SAP services. Novell network numbers must still be assigned to the appropriate interfaces with the **novell network** subcommand.

Note: If you plan to use DECnet and Novell routing concurrently on the same interface, you should enable DECnet routing first, then enable Novell routing without specifying the optional MAC address. If you do this in the reverse order (that is, enable Novell, then DECnet), routing for Novell will be disrupted.

Use the **no novell routing** command to disable Novell IPX routing.

Enabling Novell on an Interface

To enable Novell routing on a particular interface, use the **novell network** interface subcommand. The full syntax of this command follows.

novell network *number*
no novell network *number*

The argument *number* is the number of the Novell network to which that interface is attached. Novell packets received on an interface which do not have a Novell network number are ignored. Use the **no novell network** command with the network number to disable Novell on the interface.

Example:

```
novell network 2f
```

Repairing Corrupted Network Numbers

In some early implementations of Novell client software, it was possible for the client's network number to be corrupted. The **novell source-network-update** interface subcommand repairs corrupted network numbers by setting the source network field of any packet with a hop count of zero to the local network number. The full syntax of this command follows.

novell source-network-update
no novell source-network-update

The route cache must be disabled or this command will not work, and this is done using the **no novell source-network-update** command.

Note: This command will interfere with the proper working of OS/2 Requestors. Do not use this command in a network where OS/2 Requestors are present.

Example:

```
novell network 106A
novell source-network-update
no novell route-cache
```

Novell Encapsulation

There are two different data formats used by Novell on Ethernets. Use the **novell encapsulation** interface subcommand to select which data format or encapsulation is used on an Ethernet interface.

novell encapsulation *keyword*

The default keyword argument is **novell-ether**, which specifies Novell IPX over Ethernet using Novell's variant of IEEE 802.2 encapsulation. The keyword **arpa** is used when the Novell systems must communicate with other vendors' systems, such as DEC VAX/VMS. In this case, Ethernet-style encapsulation is used with a protocol type of 8137.

Note: On Token Rings, only one style of encapsulation exists. Some Novell nodes do not recognize Token Ring packets with the source-route bridging RIF field set. You can work around this Novell discrepancy by using the **no multiring** interface subcommand on Token Ring interfaces that are used for Novell IPX routing. See the chapter "Configuring Source-Route Bridging" for more information.

Configuring Static Routes

Static routes for a Novell network can be specified with the **novell route** global configuration command. The full syntax of the command follows.

novell route *network network.address*
no novell route *network network.address*

The **novell route** command causes packets received for the specified network to be forwarded to the specified router, whether or not that router is sending out dynamic routing.

Use the **no novell route** command with the appropriate arguments to remove the route.

Example:

If the router that handled traffic for network 5e had the address, *3abc.0000.0c00.1ac9*, then you would enter this command:

```
novell route 5e 3abc.0000.0c00.1ac9
```

Note: Be careful when assigning static routes. When links associated with static routes are lost, traffic may stop being forwarded, although alternative paths are available.

Setting Maximum Paths

To set the maximum number of multiple paths that the router will remember and use, use the **novell maximum-paths** global configuration command. The command was designed to increase throughput by using multiple paths. It remembers higher bandwidth routes in preference to lower bandwidth routes. The full syntax of the command follows:

novell maximum-paths *paths*
no novell maximum-paths

The argument *paths* is the number of paths to be remembered. For a given destination, multiple paths of equal cost will be remembered. The default value for *paths* is 1. Output will be determined in round-robin fashion over these multiple paths at the packet level.

The **no novell maximum-paths** command restores the default.

The EXEC command **show novell routes** displays these additional routes and the maximum path value.

Setting Novell Update Timers

To allow the Novell routing update timers to be set on a per-interface basis, use the **novell update-time** interface subcommand. Full syntax follows.

novell update-time *seconds*
no novell update-time

Internal Novell timers are affected by the value set for the *seconds* argument, as follows:

- Novell routes are marked invalid if no routing updates are heard within six times the value of the update timer and are advertised with a metric of infinity.
- Novell routes are removed from the routing table if no routing updates are heard within eight times the value of the update timer.
- The default value for the **update-time** *seconds* argument is 60.
- The granularity of the update timer is determined by the lowest value defined.
- The minimum is ten seconds.

The **no novell update-time** command restores the default of 30 seconds.

Example:

In the example listed below, the granularity would be 20 because that is the lowest value specified for that protocol.

```
interface serial 0
novell update-time 40
interface ethernet 0
novell update-time 20
interface ethernet 1
novell update-time 25
```

Note: Be careful when using this command. It can be used only in an all-Cisco environment, and all timers should be the same for routers connected to the same network segment.

The EXEC command **show novell interface** displays the value of these timers.

Filtering Novell Packets

Cisco's implementation of the Novell IPX software provides three types of filtering:

- Access lists, or packet filtering, which controls whether or not packets are sent out the filtered interface.
- Routing update filtering, which controls to which Novell IPX networks the router advertises routes to, depending on the type used.
- SAP filtering, which controls the services the router knows about, and which services the router advertises.

In setting up these packet filters, care must be taken to not set up filtering conditions that result in packets falling through a "black hole." This can happen, as an example, when the software is configured to advertise services on a network with access lists configured to deny these packets, or when a network is configured to advertise services on a network that is unreachable because routing updates are filtered out by routing update filtering.

Keep these pitfalls in mind while configuring the filter types, each discussed in the following sections.

Configuring Novell IPX Access Lists

Simple Novell IPX access lists are numbered from 800 to 899 and filter on the source and destination addresses only.

The command syntax for standard Novell IPX access lists is lengthy. For typographic reasons, the command example is shown on multiple lines; it must be on a single line when given as a configuration command. The full command syntax for the Novell **access-list** global configuration command follows.

access-list *number* {**deny**|**permit**} *novell-source-network*[[.*source-address*[*source-mask*]] *novell-destination-network*[.*destination-address* [*destination-mask*]]
no access-list *number*

The only required argument for standard Novell IPX access lists is the Novell IPX source network. The rest of the parameters are optional except that the source and/or destination address masks are present only if the corresponding source and/or destination address was entered.

Use the **no access-list** command with the appropriate access list number to remove the access list.

Example:

The following example denies access from source network –1 (all Novell IPX networks) to destination network 2.

```
access-list 800 deny -1 2
```

The following example denies access from Novell IPX source address *0000.0c00.1111*.

```
access-list 800 deny 1.0000.0c00.1111
```

The following example denies access from all nodes on network 1 that have a source address beginning with *0000.0c*.

```
access-list 800 deny 1.0000.0c00.1111 0000.00ff.ffff
```

The following example denies access from source address *1111.1111.1111* on network 1 to destination address *2222.2222.2222* on network 2.

```
access-list 800 deny 1.1111.1111.1111 0000.0000.0000 2.2222.2222.2222
0000.0000.0000
```

Configuring Extended Novell Access Lists

Extended Novell IPX access lists filter on protocol information as well; numbers for the extended lists range from 900 to 999. The command syntax for extended Novell IPX access lists is again rather lengthy as a configuration command (that must be typed on one line):

access-list *number* {**deny**|**permit**} *novell-protocol source-network*.[*source-address* [*source-mask*]] *source-socket destination-network*.[*destination-address* [*destination-mask*]] *destination-socket*
no access-list *number*

The source and destination addresses and masks are optional. The protocol number *novell-protocol* is the only required parameter. A network number of –1 matches all networks; a socket number of 0 matches all sockets.

Use the **no access-list** command with the appropriate access list number to remove the access list.

Example:

The following example denies access to protocol *1* from source network *1*, source socket *1234* to destination network *2*, destination socket *1234*.

```
access-list 900 deny 1 1 1234 2 1234
```

The following example illustrates the use of all possible parameters:

```
access-list 900 deny 1 1.1111.1111.1111 0000.0000.0000 1234
  2.2222.2222.2222 0000.0000.0000 1234
```

Filtering Outgoing Traffic

The Novell IPX access list group number is assigned with the **novell access-group** interface subcommand. The syntax for this command follows:

novell access-group *number*
no novell access-group *number*

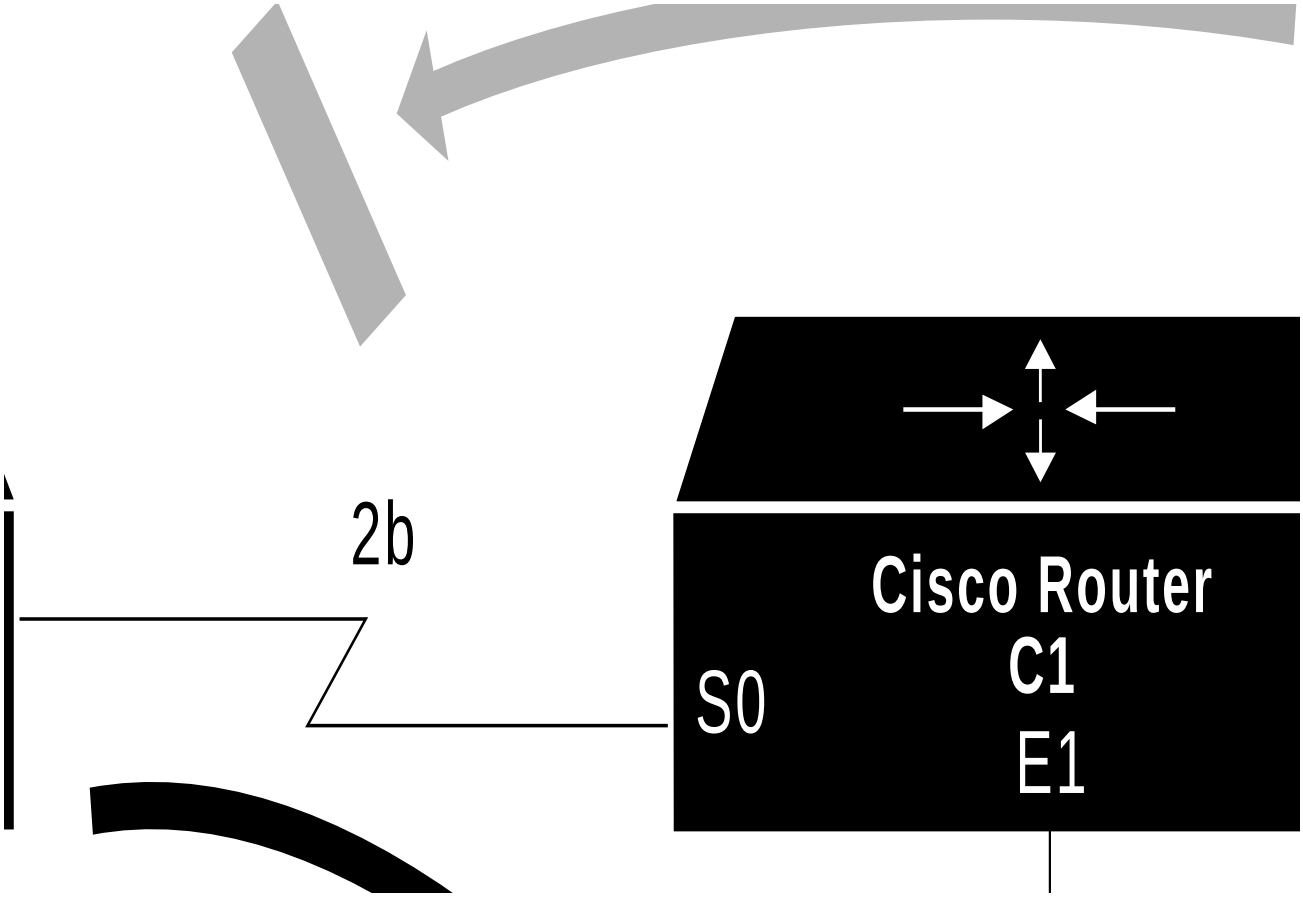
The argument *number* refers to the appropriate Novell access list number. All outgoing packets forwarded through the interface will be filtered by this access list.

Use the **no novell access-group** command with the appropriate group number to remove the access list group number from the interface.

Example of Controlling Traffic with Access Lists

Using access lists to manage traffic routing can be a powerful tool in overall network control. However, it does require a certain amount of planning and the appropriate application of several related commands. Figure 1-1 illustrates a network featuring two Cisco routers connecting a number of network segments.

Figure 1-1 Multiple Novell Servers Requiring Access Control



For the purposes of illustrating access control, the network in Figure 1-1 has the following specific requirements:

- Resources on networks *3c* and *4d* are to be allowed free access to each other through router *C1*.
- Resources on network *4d* are to be allowed access to resources on network *aa*.
- Resources on network segment *3c* are not allowed access resources on *aa*.
- Resources on network *aa* are not allowed to access resources on *3c*.
- All networks can access resources on segment *4d*.

The configuration for this environment can be defined using simple access lists as illustrated in the following example. In this example, the global configuration command **access-list** and interface subcommands **novell network** and **novell access-group** are applied to router/bridge *C1* in Figure 1-1.

Note: The commands in configuration files are executed in a sequential, first match, top down basis. Plan the placement of command lines carefully, especially when specifying and applying access lists. Access lists are always applied to the *transmitting* interface.

Example 1:

This first example is applied to router *C1*, permitting resources on network *4d* to access resources on network aa. It implicitly denies all other traffic.

```
access-list 800 permit 4d aa
```

If you want to explicitly deny all other traffic, you add the following line:

```
access-list 800 deny -1 -1
```

Example 2:

This example assigns network number *2b* to the first serial interface, then applies access group 800 and permit resources on network *4d* to access resources on network *3c*. Again, you explicitly deny all other traffic.

```
interface serial 0
novell network 2b
novell access-group 800
access-list 800 permit 4d 3a
access-list 801 deny -1 -1
```

Example 3:

This example assigns a network number and access group 801 to interface Ethernet 1, and applies a network number to the second Ethernet address. There are no explicit permissions or denials for interface Ethernet 1.

```
access-list 801 permit 4d 3c
interface ethernet 0
novell network 3c
novell access-group 801
interface ethernet 1
novell network 4d
```

Note: This configuration example does not address access control for segment cc on router *C1*. However, given no other restrictions or specific permissions, it implies that resources on cc can access resources on *4d*, but cannot access resources on *3c*. In addition, resources on *4d* cannot access resources on network *cc*.

Filtering Novell Routing Updates

This section describes the filtering commands that use access lists to control which routing information is accepted, or passed on, within Novell networks. The commands filter incoming traffic, outgoing routing updates, and specific routers.

Each access list entry contains only one address parameter. How this address is interpreted is defined by the command that will use the list.

As with all other Cisco access lists, an implicit *deny everything* is defined at the end of the list. If this is not desired, an explicit *permit everything* definition must be included at the end of the list.

Each filter type is described in the following sections.

Establishing Input Filters

To control which networks are added to the routing table, use this interface subcommand:

novell input-network-filter *access-list-number*

The argument *access-list-number* is the access list number specified in the **novell access-list** command.

Example:

In the following example, access list 876 controls which networks are added to the routing table when Novell routing updates are received. The address in the access list is a source network.

```
access-list 876 permit 1b
interface ethernet 1
novell input-network-filter 876
```

This configuration causes network *1b* to be the only network that is accepted from updates received on the defined Ethernet interface.

Establishing Output Filters

To control the list of networks that are sent in routing updates, use the interface subcommand:

novell output-network-filter *access-list-number*

The argument *access-list-number* is the access list number specified in the **novell access-list** command.

Example:

In the following example, access list 896 controls which networks are sent out in routing updates. The address parameter is the desired network.

```
access-list 896 permit 2b
interface serial 1
novell output-network-filter 896
```

This configuration causes network *2b* to be the only network advertised in Novell routing updates sent on the defined serial interface.

Establishing Router Filters

To control the list of routers from which data will be accepted, use this interface subcommand:

novell router-filter *access-list-number*

The argument *access-list-number* is the access list number specified in the **novell access-list** command.

Example:

In this example, access list 866 controls from which router data will be accepted. In this case, the address parameter is the address of the router.

```
access-list 866 permit 3c.0000.000c0.047d
interface serial 0
novell router-filter 866
```

Information from a disallowed router is ignored.

Building SAP Filters

A common source of traffic on Novell networks is the SAP-based messages generated by Novell servers and Cisco routers as they broadcast their available capabilities. Control of SAP messages can be established with Cisco routers using several facilities. Access lists and SAP filters combine to allow you to control how SAP messages from network segments or specific servers are routed among Novell networks.

Defining Access Lists for SAP Filtering

To define an access list for filtering SAP requests, use this variation of the **access-list** command:

access-list *number* **permit**|**deny** *network*.[*address*] [*service-type*]

The argument *number* is the SAP access list, which must is a decimal number in the range 1000 to 1099.

Enter the keyword **permit** or **deny** to establish the type of access desired. Permit or deny access is based on the data provided.

The argument *network* is a hexadecimal Novell network number; 0 defines the local network, –1 defines all networks.

The optional *address* argument is a Novell node address.

The *service-type* argument defines the service type to filter; 0 is all services. Service types are entered in hexadecimal. Examples of the service types that may be entered are listed in Table 1-1.

Table 1-1 Sample Novell SAP Services

Description	Service Type
Unknown	0
User	1
User Group	2
Print Queue	3
File Server	4
Job Server	5
Gateway	6
Print Server	7
Archive Queue	8
Archive Server	9
Job Queue	A
Administration	B
Remote Bridge Server	24
Advertizing Printer Server	47
Wildcard	blank (no entry)

Example—Service Type Specification

```
! Deny access from all nets for service 4:
access-list 1001 deny -1 4
! Permit access from all nets to all other services:
access-list 1001 permit -1
```

Note: In the example outlined above, companion interface specification must be defined for this access list to be applied. Once applied to an interface, this access list definition blocks all access to service type 4 (file service) on directly attached network by resources on other Novell networks. The second specification explicitly allows access to all other available services on the interface.

Configuring Novell SAP Filters

Use these commands to filter Novell SAP messages.

novell input-sap-filter *access-list-number*
novell output-sap-filter *access-list-number*
novell router-sap-filter *access-list-number*

These commands take a SAP Novell access list number as their input. The range for SAP lists is 1000 to 1099.

Follow these guidelines to use SAP filtering:

- When the **novell input-sap-filter** list is enabled, use the list to determine the services that will be accepted.
- When the **novell output-sap-filter list** is enabled, use the list to determine the services that will be included in SAP updates from Cisco routers.
- When the **novell router-sap-filter** list is enabled, use the list to determine the router from which the Cisco router will hear SAP messages, and the service type.

Example SAP Input Filter

Input SAP filters are applied prior to a Cisco router accepting information about a service. In the example that follows, Cisco router *C1* (illustrated in Figure 1-2FigureFigure 1-2 1-2) will not accept and, consequently not advertise, any information about Novell server *F*. However, *C1* will accept information about all other servers on the network *3c*. Cisco router *C2* will receive information about servers *D* and *B* in this example.

Example:

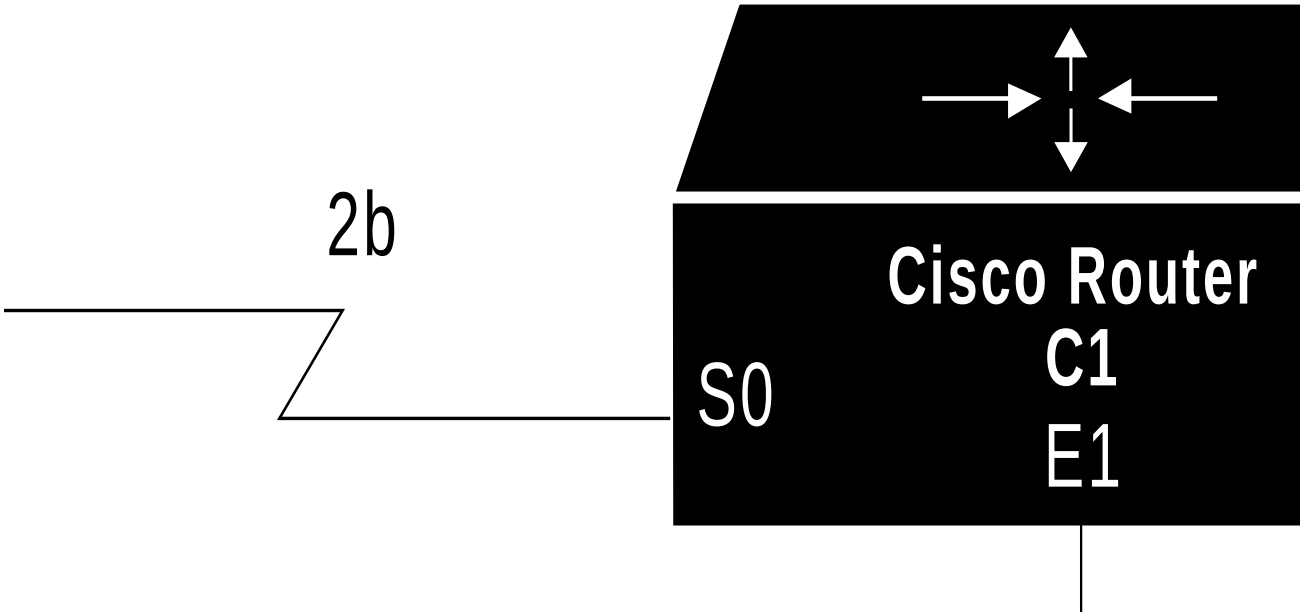
This example configures router *C1*. The first line denies server *F*. It accepts all other servers.

```
access-list 1000 deny 3c.0800.89a1.1527
access-list 1000 permit -1
interface ethernet 0
novell network 3c
novell input-sap-filter 1000
interface ethernet 1
novell network 4d
```

```
interface serial 0
novell network 2b
```

Note: Please note that NetWare 386 servers use an internal network and node number as their address for access list commands (the first configuration command in this example).

Figure 1-2 SAP Input Filter



Example SAP Output Filter

Output SAP filters are applied prior to a Cisco router sending information out a specific interface. In the example that follows, Cisco router *C1* (illustrated in Figure 1-2) is prevented from advertising information about Novell server *A* out interface Ethernet 1, but can advertise server *A* on network *3c*.

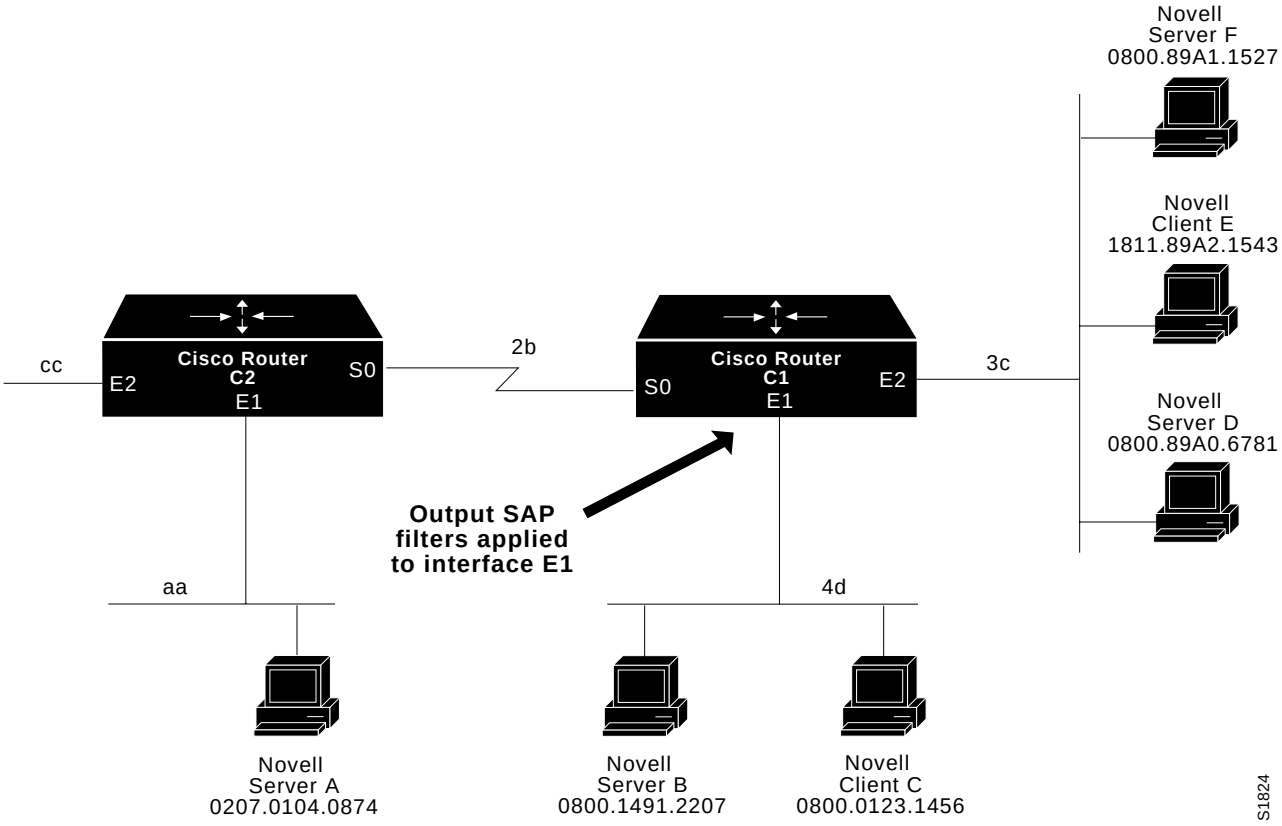
Example:

The following example refers to router *C1*. The first line denies server *A*. All other servers

are permitted.

```
access-list 1000 deny aa.0207.0104.0874
access-list 1000 permit -1
interface ethernet 0
novell net 3c
interface ethernet 1
novell network 4d
novell output-sap-filter 1000
interface serial 0
novell network 2b
```

Figure 1-3 SAP Output Filter



Novell Broadcast Helper Facilities

Cisco's helper facilities provide a flexible set of tools to help you manage Novell network broadcast traffic. Several configuration options allow network administrators to tailor the way broadcasts generated by Novell clients are forwarded through a network.

If Novell clients and servers are attached to the same network segment, this basic function (blocking broadcasts) is acceptable and often preferred, since it helps reduce unwanted traffic

among networks. However, when clients must broadcast through a router to a remotely-located server, several modifications to the Cisco system configuration are required. To make these modifications, use the Novell helper functions.

Cisco routers support flooding. *Flooding*, as the name suggests, forwards broadcasts to all networks.

The key to controlling Novell broadcasts (rather than simply blocking them) rests with the use of several commands specific to Cisco's Novell IPX routing implementation:

- **novell helper-address**—Explicitly specifies the address of a target Novell server (or network) to which broadcast packets are sent (applied to a specific interface).
- **novell helper-list**—Assigns access lists to interfaces to control broadcast traffic (applied to a specific interface).
- **access-list**—A general Cisco traffic-controlling tool that can be tailored to help manage broadcast traffic in a Novell network environment.

Defining a Helper List

The **novell helper-address** and **novell helper-list** interface subcommands are defined briefly below, while the global configuration command **access-list** is described in a preceding section. Following the "helper" facility definitions, several typical applications illustrate how to use Cisco's helper and access list mechanisms together within the context of Novell-based internetworking environments.

The **novell helper-list** interface subcommand specifies that only those packets that pass the specified Novell access list are forwarded to a remote Novell server. (The only exception to this rule is that all-nets flooded broadcasts (our next topic) and NetBIOS are ALWAYS forwarded, regardless of how you set the helper-list command.) The syntax for this command is:

novell helper-list *access-list-number*

The argument *access-list-number* specifies the access list. The network numbers in that list are expressed in hexadecimal values.

Note: Since the destination address of a broadcast is by definition the broadcast address, this is only useful for filtering based on the source of the broadcast packet. This can be used to prevent nodes from discovering services they should not use.

Specifying Target Novell Servers

To forward broadcast packets that match the access list specified by the **novell helper-list** subcommand, use the **novell helper-address** interface subcommand:

novell helper-address *net.host*

This subcommand causes all-nets broadcasts to be forwarded to *net.host*. The argument

net.host is a dotted combination of the network and host addresses as explained in the **novell route** subcommand.

Incoming unrecognized broadcast packets that match the access list will be forwarded on to the address specified by the argument *net.host*.

The Cisco routers support all network (*all nets*) flooding. To configure the all nets broadcast flooding, define the Novell helper address for an interface as:

```
-1.FFFF.FFFF.FFFF
```

On systems configured for Novell routing, this helper address will be displayed as:

```
FFFFFFFF:FFFF.FFFF.FFFF
```

Note: Although care has been taken to keep traffic to a minimum, some duplicates will be unavoidable. Under certain conditions (where loops exist) flooding can propagate bursts of excess traffic that will, eventually age out, when the hop count reaches its limit. Use the broadcast address carefully and only when necessary.

Using Helper Facilities to Control Broadcasts

Use of the **helper-list** and **helper-address** tools is best illustrated with examples. The following illustrations and accompanying descriptions outline the application of access lists, helper lists, and helper addresses to forward traffic to a specific network or to a node, and to flood broadcast messages on all attached links.

Forwarding to an Address

You can direct broadcasts to a specific network or host (node) on a segment. The following examples illustrate both these forwarding options.

Figure 1-4 shows a Cisco router (C1) connected to several Ethernets. In this environment, all Novell clients are attached to segment aa, while all servers are attached to segment bb. In controlling broadcasts, the following conditions are to be applied:

- Only type 10 broadcasts are to be forwarded.
- The Novell clients on network aa are to be allowed to broadcast to any server on network *bb.*
- All-nets broadcasts are always flooded.

Figure 1-4 SAP Output Filter

Allows type 10 broadcasts only

Interfaces E1 and E2 do not require application of any specific permissions to meet the conditions for this example, since broadcasts are by default blocked by the router.

Example:

This example configures the router shown in Figure 16-4. The first line permits traffic of type 10 from network *aa*. Then the interface and network commands configure a specific interface. The helper-address command permits broadcast forwarding from Network *aa* to *bb*. The last line forwards type 10 broadcasts from networks *aa* to *bb*.

```
access-list 900 permit 10 aa
interface ethernet 0
novell network aa
novell helper-address bb.ffff.ffff.ffff
novell helper-list 900
```

Any downstream network that is cascaded beyond network aa (for example, some arbitrary network *aa1*) will not be able to broadcast to network *bb* through router *C1*, unless the routers partitioning networks *aa* and *aa1* are configured to forward broadcasts with a series of configuration entries analogous to the example provided for Figure 16-4. These entries must be applied to the input interface and be set to forward broadcasts between directly connected networks. In this way, traffic can be passed along, in a directed manner, from network to network.

Example:

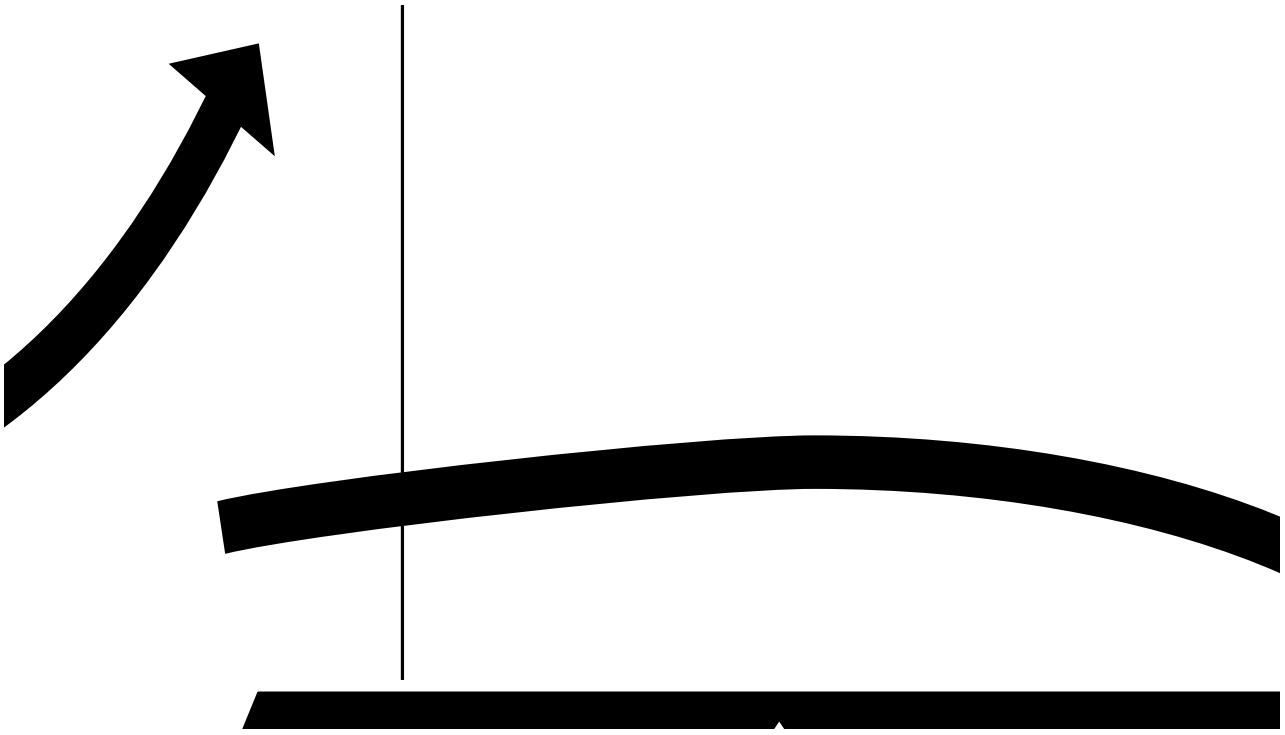
The example provided below rewrites the **novell helper-address** command line to direct broadcasts to server *A* in Figure 16-4.

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

Forwarding to All Networks

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

Figure 1-5 Type 10 Broadcast Flooding



As with the prior example, the configuration for this environment can be defined using an extended access list, a helper list and a helper address.

Example:

In this example, the first line permits traffic of type 10 to network *2b1*. Then the first Ethernet interface is configured with a network number. The helper address is defined and the helper list limits forwarding to type 10 traffic.

```
access-list 901 permit 10 2b1
interface ethernet 0
novell network 2b1
novell helper-address -1.ffff.ffff.ffff
novell helper-list 901
interface ethernet 1
novell network 3c2
interface ethernet 2
novell network 4a1
```

```
interface ethernet 3
novell network 5bb
```

In this example, type 10 broadcasts from network *2b1* are forwarded to all directly connected networks. All other broadcasts are blocked. If all broadcasts are to be permitted, delete the **novell helper-list** entry.

Enabling Novell Fast Switching

Novell fast switching allows higher throughput by switching the packet using a cache created by previous transit packets. Fast switching also provides load sharing on a per-packet basis.

Use the **novell route-cache** interface subcommand to enable fast switching. The full syntax for this command follows.

novell route-cache
no novell route-cache

When Novell routing is enabled, by default, Novell fast switching is enabled on the appropriate interface. Use the **no novell route-cache** command to disable fast switching.

Restricting SAP Updates

To configure less frequent SAP updates over slow links, use the interface subcommand **novell sap-interval**. This command has the following syntax:

novell sap-interval *interval*

Use the *interval* argument to set the interval between SAP updates. If *interval* is zero, periodic updates are not sent. A message is sent only when the server first appears and when it goes down. The default value for the argument *interval* is one minute. This is the value used by the Novell servers.

Example:

In this example, SAP updates are sent (and expected) on interface serial 0 every five minutes.

```
interface serial 0
novell sap-interval 5
```

All Novell servers and routers on a particular network require the same SAP interval or they are likely to decide that a server is down, even though it is actually up. Since it is impossible to change this value on most PC-based servers, you should never change the interval for an Ethernet or Token Ring that has actual servers on it. This subcommand is most useful on limited bandwidth point-to-point links or X.25 interfaces, where one prefers to use as little bandwidth as possible for sending the SAP updates.

Note: Setting the interval to zero is especially dangerous. Routers that were inaccessible for any reason at the time of a server power up or shut down will miss that event, and will either fail to learn about the existence of new servers, or will fail to detect the server shut down.

SAP Update Delays

Some slow Novell servers lose SAP updates because they cannot keep up the same brisk processing pace as the Cisco routers can. The **novell output-sap-delay** interface subcommand allows you to set a delay between SAP updates, in effect forcing the Cisco router interface to pace its output to the slower-processing needs of the Novell servers. If your server is not slow and is not losing SAP updates, you can skip this configuration command. The full syntax of the command follows:

novell output-sap-delay *delay*
no novell output-sap-delay

The parameter *delay* is measured in milliseconds.

The **no novell output-sap-delay** disables the delay mechanism.

Example:

```
novell network 106A
novell output-sap-delay 200
```

Novell Configuration Example

The following configuration commands enable Novell routing, defaulting the Novell host address to that of the first IEEE-conformance (no serial, for example) interface. Routing is then enabled on Ethernet 0 and Ethernet 1 for Novell networks *2abc* and *1def*, respectively.

Example:

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

Note: The network numbers used must match the numbers used by Novell file servers on the same cable.

Monitoring the Novell IPX Network

Use the EXEC commands described in this section to obtain displays of activity on the Novell IPX network.

Displaying the Novell Cache Entries

Use the **show novell cache** command to display a list of fast-switching cache entries. Enter this command at the EXEC prompt:

show novell cache

Following is sample output:

```
Novell routing cache version is 9
Destination                Interface                  MAC Header
*1006A                 Ethernet0           00000C0062E600000C003EB0064
*14BB                  Ethernet 1          00000C003E2A00000C003EB0064
```

In the sample display, valid entries are marked by an asterisk (*).

Displaying Novell Interface Parameters

Use the **show novell interface** command to display the Novell parameters that have been configured on the interfaces. Enter this command at the EXEC prompt:

show novell interface [*interface unit*]

An optional interface name can be specified with the *interface unit* arguments to see information for a specific interface. Following is sample output:

```
Ethernet 0 is up, line protocol is up
  Novell encapsulation is NOVELL-ETHER
  Novell address is 1006A.0000.0c00.62e6
  Outgoing access list is not set
  Novell SAP update interval is 1 minute(s)
  Novell Helper access list is not set
  SAP Input filter list is not set
  SAP Output filter list is not set
  SAP Router filter list is not set
  Input filter list is not set
  Output filter list is not set
  Router filter list is not set
  Update time is 30 seconds
  NOVELL Fast switching enabled
```

Displaying the Novell Routing Table

Use the **show novell route** EXEC command to display the Novell routing table. Enter this command at the EXEC prompt:

show novell route

Following is sample output:

```
Codes: R - RIP derived, C - connected, S - static, 2 learned routes

Maximum allowed path(s) are/is 1
R Net 1001 [1/1] via 1006.aa00.0400.6508,  94 sec, 0 uses, Ethernet0
R Net 1003 [1/1] via 1006.aa00.0400.6508,  94 sec, 0 uses, Ethernet0
C Net 13A is directly connected, 0 uses, Ethernet1 (down)
C Net 1006A is directly connected, 0 uses, Ethernet0
```

In the display, the leading character R indicates routes learned via RIP, C indicates connected entries, and S indicates statically defined entries. The square brackets contain the metric/delay field reports. The first number is the metric used to make a routing decision; the second number is the delay expressed in IBM clock ticks. The field is not used to make routing decisions, however, the Novell server will use this field to decide which of two equal metric routes to use, thereby giving it a tie-breaking function.

Displaying Novell Servers

Use the **show novell servers** EXEC command to list the servers discovered through SAP advertisements. Enter this command at the EXEC prompt:

show novell servers

Following is sample output:

```
Type       Name         Net Address               Port          Hops

4        SYSOP        2a.0206.00a2.41ec:0450       2          Ethernet2
4        MKTG         3c.0800.00a3.45ef:0450       2          Ethernet5
4        SERVICE      4d.0a44.008a.0220:0450
Ethernet10
4        FINANCE      1a.0080.a246.0001:0450
Ethernet10
```

For each known server in the network, the display lists its name, complete address, number of hops distant it is and the interface through which it can be accessed (through which it was discovered).

Displaying Novell Traffic

Use the **show novell traffic** EXEC command to display information on the number and type of Novell packets transmitted and received. Enter this command at the EXEC prompt:

show novell traffic

Following is sample output:

```
Rcvd:  68112 total, 0 format errors, 0 checksum errors, 0 bad hop count,
        68102 local destination, 0 multicast
Bcast:  68102 received, 43745 sent
Sent:   43745 generated, 0 forwarded
        0 encapsulation failed, 10 no route
```

```
SAP:    0 SAP requests, 0 SAP replies
        0 SAP advertisements received, 0 sent
 Echo: Rcvd 0 requests, 0 replies
       Sent 0 requests, 0 replies
        0 unknown
```

The following notes apply to the less-obvious statistics in this screen:

- Format errors are reported whenever a bad packet is detected (for example, a corrupted header).
- Checksum errors should not be reported, since IPX does not use a checksum.
- Bad hop count increments when a packets hop count exceeds 16.
- Encapsulation failed is registered when the router is unable to encapsulate a packet.
- The unknown counter increments when packets are encountered that the router is unable to forward (for example, misconfigured helper-address, or no route available).

Novell Ping Command

To execute the **ping** command on a Cisco network server configured for Novell routing, enter **novell** at the **ping** protocol prompt and the Novell routing address at the address parameter prompt. The defaults are enclosed in brackets at each prompt.

Here is a sample:

```
Protocol [ip]: novell
Target Novell Address: 1006A.0000.0c00.62e6
Repeat Count [5]:
Datagram Size [100]:
Timeout in seconds [2]:
Verbose [n]:
Type escape sequence to abort.
Sending 5 100-byte Novell echoes to 1006A,0000,0c00,62e6, timeout is 2
seconds.
!!!!!!!
Success rate is 100%, round trip min/avg/max  = 1/2/4 ms.
```

See the section “Testing Connectivity with the Ping Command” in the chapter “Managing the System” for more information.

Note: This command only works on Cisco network servers running Release 8.2 (or later) software. Novell devices will not respond to this command.

Debugging the Novell IPX Network

Use the commands described in this section to troubleshoot and monitor the Novell IPX network. For each **debug** command, these is a corresponding **undebug** command that turns off message logging.

debug novell-packet

The **debug novell-packet** command outputs information about packets received, transmitted, and forwarded.

debug novell-routing

The **debug novell-routing** command prints out information on Novell routing packets.

debug novell-routing events

The **debug novell-routing events** command provides a subset of the information displayed by the **debug novell-routing** command.

debug novell-sap

The **debug novell-sap** command displays additional information about Novell Service Advertisement packets.

debug novell-sap-events

The **debug novell-sap-events** command provides a subset of the information displayed by the **debug novell-sap** command.

Novell IPX Global Configuration Command Summary

The following is an alphabetical list of the Novell IPX global configuration commands. These commands specify system-wide parameters for Novell IPX support.

[**no**] **novell routing** [*host-address*]

Enables and disables Novell routing and Novell RIP routing and SAP services. You can also use this command to specify the system-wide host address to use with the optional argument *host-address*. If you do not specify an address, the MAC address of the first Ethernet, Token Ring, or FDDI interface is used. If there are no satisfactory interfaces

present, you must specify the host address argument. The address must not be multicast. Assign Novell network numbers to the appropriate interfaces with the **novell network** subcommand.

[**no**] **novell route** *network network.address*

Specifies or removes static routes for a Novell network. When specified, the command causes packets received for the specified network to be forwarded to the specified router, whether or not that router is sending out dynamic routing.

[**no**] **novell maximum-paths** *paths*

Sets the maximum number of multiple paths that the router will remember and use. The argument *paths* is the number of paths to be remembered. The **no** form of the command restores the default.

[**no**] **access-list** *number* **deny**| **permit** *novell-source-network*[.*source-address*] *source-mask novell-destination-network.destination-address destination-mask*

Specifies standard Novell IPX access lists. Standard Novell IPX access lists are numbered from 800 to 899 and filter on the source and destination addresses only. An access list command must be completely specified on a single line when given as a configuration command. The only required parameter for standard Novell IPX access lists is the Novell IPX source network. The rest of the parameters are optional except that the source and/or destination address masks are present only if the corresponding source and/or destination address was entered. The **no** form of the command removes any access list in the current image with the specified number.

[**no**] **access-list** *number* **deny**|**permit** *novell-protocol source-network*.[*source-address* [*source-mask*]] *source-socket destination-network*.[*destination-address* [*destination-mask*]]*destination-socket*

Specifies extended Novell IPX access lists. The source and destination addresses and masks are optional. The protocol number *novell-protocol* is the only required parameter. A network number of –1 matches all networks; a socket number of 0 matches all sockets. Extended Novell IPX access lists filter on protocol information as well; numbers for the extended lists range from 900 to 999. The **no** form of the command removes any access list in the current image with the specified number.

[**no**] **access-list** *number* **permit**|**deny** *network*.[*address*] [*service-type*]

Defines an access list for filtering SAP requests. The argument *number* is a decimal number in the range of 1000 to 1099. Enter the keyword **permit** or **deny** to establish the type of access desired. Permit or deny access is based on the data provided. The argument *network* is a hexadecimal Novell network number; 0 defines the local network, –1 defines all networks. The optional *address* argument is a Novell host address. The *service-type* argument defines the service type to filter; 0 is all services. Service types are entered in hexadecimal.

novell encapsulation *keyword*

Selects which data format or encapsulation is used on an Ethernet interface. The default keyword argument is **novell-ether** which specifies Novell IPX over Ethernet using Novell's variant of IEEE 802.2 encapsulation. The keyword **arpa** is used when the Novell systems must communicate with other vendors' systems, such as DEC VAX/VMS. In this case, Ethernet-style encapsulation is used with a protocol type of 8137.

[**no**] **novell input-sap-filter** *access-list-number*
[**no**] **novell output-sap-filter** *access-list-number*
[**no**] **novell router-sap-filter** *access-list-number*

Configure Cisco routers to filter the acceptable source of Novell SAP messages; the intended destination of SAP messages; or the specific router from which SAP filters will be accepted. These commands take a SAP Novell access list number as their input. The range for SAP lists is 1000 to 1099. The **no** forms of the commands remove the filters.

Novell IPX Interface Subcommand Summary

The following is an alphabetical list of the Novell IPX interface subcommands. These commands specify line-specific parameters for Novell IPX support. These subcommands must be preceded by an **interface** command.

[**no**] **novell access-group** *number*

Assigns or removes a Novell IPX access list group number to a specific interface. The argument *number* refers to the appropriate Novell access list number. All outgoing packets forwarded through the interface will be filtered by this access list.

[**no**] **novell helper-address** *net.host*

Broadcast packets that match the access list specified by the **novell helper-list** subcommand are forwarded when this command is used. This subcommand causes all-nets broadcasts to be forwarded to *net.host*. The argument *net.host* is a dotted combination of the network and host addresses as explained in the **novell route** subcommand. Incoming unrecognized broadcast packets that match the access list will be forwarded on the address specified by the argument *net.host*. This subcommand is useful for hosts which use a protocol other than SAP for advertising their availability.

[**no**] **novell helper-list** *access-list-number*

Specifies that only those packets which pass the specified Novell access list will be forwarded to the Novell helper host. The argument *access-list-number* specifies the access list. The network numbers in that list are expressed in decimal values. The **no** form of

the command disables the function.

[**no**] **novell source-network-update**

Enables the interface to provide the current network number in place of the source network number of any packet that arrives with a hop count of zero. The **no** form of the command disables the function.

[**no**] **novell input-network-filter** *access-list-number*

Explicitly specifies which networks are added to the Novell IPX routing table. The argument *access-list-number* is the access list number specified in the **novell access-list** command. The **no** form of the command disables the function.

[**no**] **novell network** *number*

Enable and disables Novell routing on a particular interface. The argument *number* is the number of the Novell network to which that interface is attached. Novell packets received on an interface which do not have a Novell network number are ignored.

[**no**] **novell output-network-filter** *access-list-number*

Explicitly specifies the list of networks that are sent in routing updates. The argument *access-list-number* is the access list number specified in the **novell access-list** command. The **no** form of the command disables the function.

[**no**] **novell output-sap-delay** *delay*

Sets the interval, measured in milliseconds, that an interface will delay, added to the usual SAP reporting interval. The **no** form of the command disables the mechanism.

[**no**] **novell route-cache**

Enables and disables Novell fast-switching. When routing is enabled, by default, Novell fast-switching is enabled on the appropriate interface. The **no** form of the command disables fast-switching.

[**no**] **novell router-filter** *access-list-number*

Specifies or removes the list of routers from which data will be accepted. The argument *access-list-number* is the access list number specified in the **novell access-list** command.

novell sap-interval *interval*

Configures less frequent SAP updates over slow links by setting the interval between SAP

updates to the number of minutes specified by the *interval* argument. If *interval* is zero, periodic updates are not sent. A message is sent only when the server first appears and when it goes down. The default value for the argument *interval* is one minute. This is the value used by the Novell servers.

[**no**] **novell update-time** *seconds*

Allows the Novell routing update timers to be set on a per-interface basis.