Novell IPX Configuration Examples

Note Effective with Cisco IOS Release 15.1(3)S, 15.2(2)T, 15.1(1)SY, and 15.2(1)E, this feature is not supported in Cisco IOS software.

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This following sections provide IPX configuration examples:

- IPX Routing Examples
- Enhanced IGRP Examples
- IPX over WAN Examples
- IPX Network Access Examples
- Helper Facilities to Control Broadcast Examples
- IPX Accounting Example

IPX Routing Examples

This section shows examples for enabling IPX routing on interfaces with a single network and with multiple networks. It also shows how to enable and disable various combinations of routing protocols.

The following sections provide these examples:

- IPX Routing on a Single Network Example
- IPX Routing on Multiple Networks Examples
- IPX Routing Protocols Examples

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

IPX Routing on a Single Network Example

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

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

IPX Routing on Multiple Networks Examples

There are two ways to enable IPX on an interface that supports multiple networks. You can use subinterfaces or primary and secondary networks. This section gives an example of each.

Subinterfaces Example

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

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

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

You can administratively shut down each of the four subinterfaces separately by using the **shutdown** interface configuration command for each subinterface. The following example shows how to administratively shut down a subinterface:

```
interface ethernet 0.3
 shutdown
```

To bring down network 1, use the following commands:

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

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

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

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

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

```
 no ipx network
interface ethernet 0.4
 no ipx network
```

Primary and Secondary Networks Example

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

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

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

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

The following example shows how to shut down network 1:

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

The following example shows how to bring the network back up:

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

The following two examples show how to shut down all four networks on the interface and remove all the networks on the interface:

```
 no ipx network
```

```
 no ipx network 1
```

The following example shows how to remove one of the secondary networks on the interface (in this case, network 2):

```
 no ipx network 2
```

The following example shows how to enable IPX routing on FDDI interfaces 0.2 and 0.3. On FDDI interface 0.2, the encapsulation type is SNAP. On FDDI interface 0.3, the encapsulation type is the Novell FDDI_RAW.

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

IPX Routing Protocols Examples

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

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

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

The following example shows how to enable RIP on networks 1 and 2 and Enhanced IGRP on network 1:

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

The following example shows how to enable RIP on network 2 and Enhanced IGRP on network 1:

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

The following example shows how to configure NLSP on two Ethernet interfaces of the router. Note that RIP is automatically enabled on both of these interfaces. This example assumes that the encapsulation type is Ethernet 802.2.

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

Enhanced IGRP Examples

The following sections show several examples of how to configure IPX Enhanced IGRP routing:

- IPX Enhanced IGRP Example
- IPX SAP-Incremental IGRP Example
- Enhanced IGRP SAP Update Examples
- Advertisement and Processing of SAP Update Examples
- IPX Enhanced IGRP Bandwidth Configuration Example

IPX Enhanced IGRP Example

The following example shows how to configure two interfaces for Enhanced IGRP routing in autonomous system 1:

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

IPX SAP-Incremental IGRP Example

The following example shows a sample configuration for enabling the IPX SAP Enhanced IGRP:

```
ipx routing
!
interface ethernet 0
 ipx network 1
 ipx sap-incremental eigrp 1
 ipx sap-incremental split-horizon
!
ipx router eigrp 100
 network 1
```

Enhanced IGRP SAP Update Examples

If an Ethernet interface has neighbors that are all configured for Enhanced IGRP, you might want to reduce the bandwidth used by SAP packets by sending SAP updates incrementally. The following example shows how to send SAP updates incrementally:

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

The following example shows how to send only incremental SAP updates on a serial line that is configured for Enhanced IGRP:

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

```
!
ipx router eigrp 1
 network 10
 network 20
```

Advertisement and Processing of SAP Update Examples

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

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

The following example shows how to configure the router to redistribute Enhanced IGRP into NLSP area1. Only services for networks 2 and 3 are accepted by the NLSP routing process.

```
access-list 1000 permit 2
access-list 1000 permit 3
access-list 1000 deny -1
!
ipx router nlsp area1
 redistribute eigrp
 distribute-sap-list 1000 in
```

IPX Enhanced IGRP Bandwidth Configuration Example

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

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

The following example shows how to configure the bandwidth of a 56-kbps circuit to 20 kbps for routing policy reasons. The Enhanced IGRP process 109 is configured to use a maximum of 200 percent (or 40 kbps) of the circuit.

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

IPX over WAN Examples

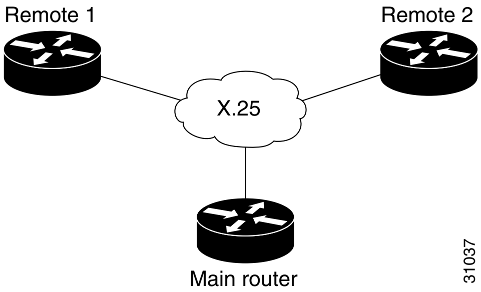
The following sections show examples of how to configure IPX over WAN and dial interfaces.

- IPX over a WAN Interface Example
- IPX over DDR Example

IPX over a WAN Interface Example

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

Figure 1 IPX over a WAN Interface



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

Configuration for Main Router

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

Configuration for Router 1

```
hostname Remote1
!
no ip routing
```

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

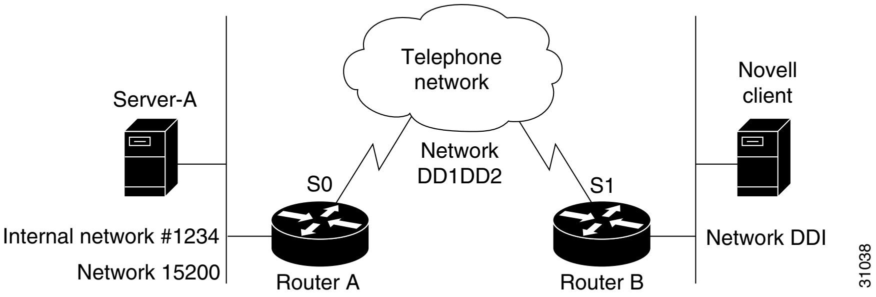
Configuration for Router 2

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

IPX over DDR Example

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

Figure 2 IPX over DDR Configuration



Routing and service information is sent every 60 seconds. The output RIP and SAP filters defined in this example filter these updates, preventing them from being sent between Router A and Router B. If you forwarded these packets, each of the two routers would need to telephone the other once every 60 seconds. On a serial link whose charges are based on the number of packets sent, this activity is generally not desirable. (This problem may not occur on a dedicated serial line.)

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

If Router A were allowed to forward the keepalive packets of the server to Router B, Router A would need to telephone Router B every 5 minutes just to send these packets. Again, on a serial link whose charges are based on the number of packets sent, this activity is generally not desirable. Instead of having Router A telephone Router B only to send keepalive packets, you can enable watchdog spoofing on Router A. The result will be that when the server connected to this router sends keepalive packets, Router A will respond on behalf of the remote client (the client connected to Router B). When SPX is used, enable spoofing of SPX keepalive packets on both routers A and B to inhibit the sending of them because both the server and the client send keepalive packets.

Use the **ipx watchdog-spoof** interface configuration command to enable and set the duration of watchdog spoofing. You can specify the number of consecutive hours spoofing is to stay enabled and the number of minutes spoofing is to stay disabled. Use this command only on a serial interface whose fast switching and autonomous switching are disabled.

The following example shows how to configure Router A. Watchdog spoofing will be enabled for 1 hour and disabled for 20 minutes, allowing the server to clean up inactive connections before being enabled again.

```
ipx routing 0000.0c04.4878
!
interface Ethernet0
     ipx network 15200
!
interface Serial0
! PPP encap for DDR(recommended)
 encapsulation ppp
 ipx network DD1DD2
! Kill all rip updates
ipx output-network-filter 801
! Kill all sap updates
ipx output-sap-filter 1001
! fast-switching off for watchdog spoofing
no ipx route-cache
! Don't listen to rip
ipx router-filter 866
! IPX watchdog spoofing
ipx watchdog-spoof 1 20
!SPX watchdog spoofing
ipx spx-spoof
! Turn on DDR
dialer in-band
 dialer idle-timeout 200
 dialer map IP 198.92.96.132 name R13 7917
 dialer map IPX DD1DD2.0000.0c03.e3c3 7917
  dialer-group 1
ppp authentication chap
! Chap authentication required
pulse-time 1
!
access-list 801 deny  FFFFFFFF
```

```
access-list 866 deny  FFFFFFFF
!  Serialization packets
access-list 900 deny  0 FFFFFFFF 0 FFFFFFFF 457
! RIP packets
access-list 900 deny  1 FFFFFFFF 453 FFFFFFFF 453
! SAP packets
access-list 900 deny  4 FFFFFFFF 452 FFFFFFFF 452
! Permit everything else
access-list 900 permit -1 FFFFFFFF 0 FFFFFFFF 0
!
access-list 1001 deny  FFFFFFFF
!
! Static ipx route for remote network
ipx route DD1 DD1DD2.0000.0c03.e3c3
!
!
! IPX will trigger the line up (9.21 and later)
dialer-list 1 list 900
```

IPX Network Access Examples

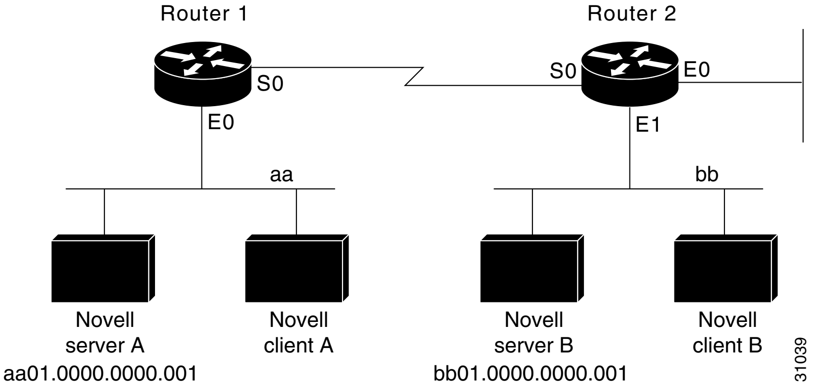
The following sections show examples of how to control access to your IPX network. The sections show the configurations for various access lists and filters.

- IPX Network Access Example
- Standard Named Access List Example
- Extended Named Access List Time Range Example
- SAP Input Filter Example
- SAP Output Filter Example
- GGS SAP Response Filter Example
- IPX NetBIOS Filter Examples

IPX Network Access Example

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

Figure 3 Novell IPX Servers Requiring Access Control



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

The following example shows how to configure Ethernet interface 1 on Router 2:

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

The following example shows how you can accomplish the same result as the previous example more efficiently by placing an input filter on interface Ethernet 0 of Router 1. You can also place the same output filter on Router 1, interface serial 0.

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

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

Logging Access Control List Violations

The following example shows how you can keep a log of all access control list violations by using the keyword **log** at the end of the **access-list** command:

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

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

The following example shows a log entry for the **access-list** command:

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

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

Standard Named Access List Example

The following example shows how to create a standard access list named fred. It denies communication with only IPX network number 5678.

```
ipx access-list standard fred
 deny 5678 any
 permit any
```

Extended Named Access List Time Range Example

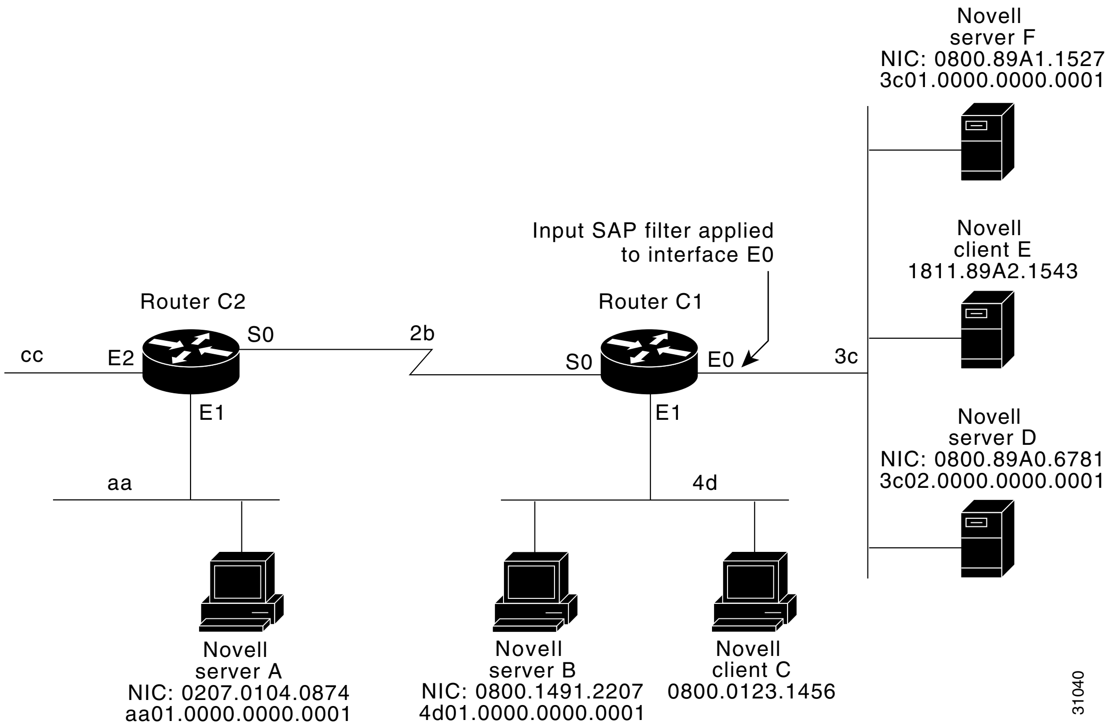
The following example shows how to create an extended access list named test. It permits SPX traffic only on Monday through Friday between the hours of 8:00 a.m. and 6:00 p.m.

```
time-range no-spx
 periodic weekdays 8:00 to 18:00
!
ipx access-list extended test
 permit spx any all any all time-range no spx
```

SAP Input Filter Example

SAP input filters allow a router to determine whether to accept information about a service.
Router C1, illustrated in Figure 4, will not accept and, consequently not advertise, any information about Novell server F. However, Router C1 will accept information about all other servers on the network 3c. Router C2 receives information about servers D and B.

Figure 4 SAP Input Filter



The following example shows how to configure Router C1. The first line denies server F, and the second line accepts all other servers.

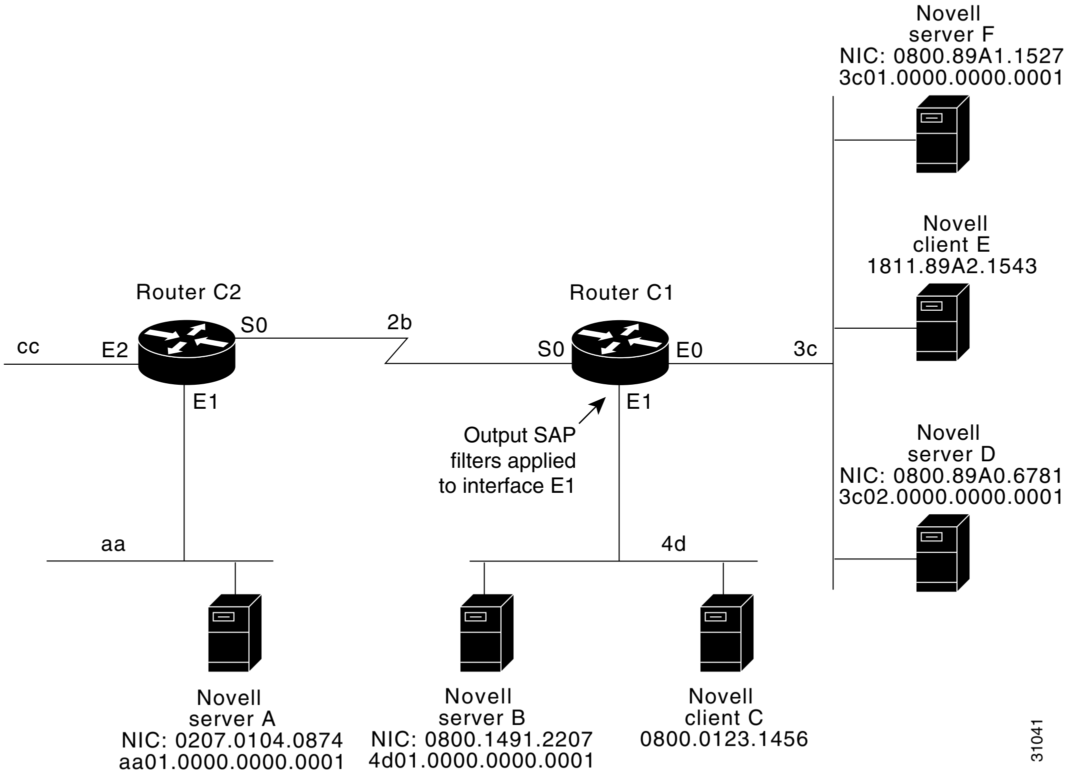
```
access-list 1000 deny 3c01.0000.0000.0001
access-list 1000 permit -1
interface ethernet 0
 ipx network 3c
 ipx input-sap-filter 1000
interface ethernet 1
 ipx network 4d
interface serial 0
 ipx network 2b
```

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

SAP Output Filter Example

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

Figure 5 SAP Output Filter



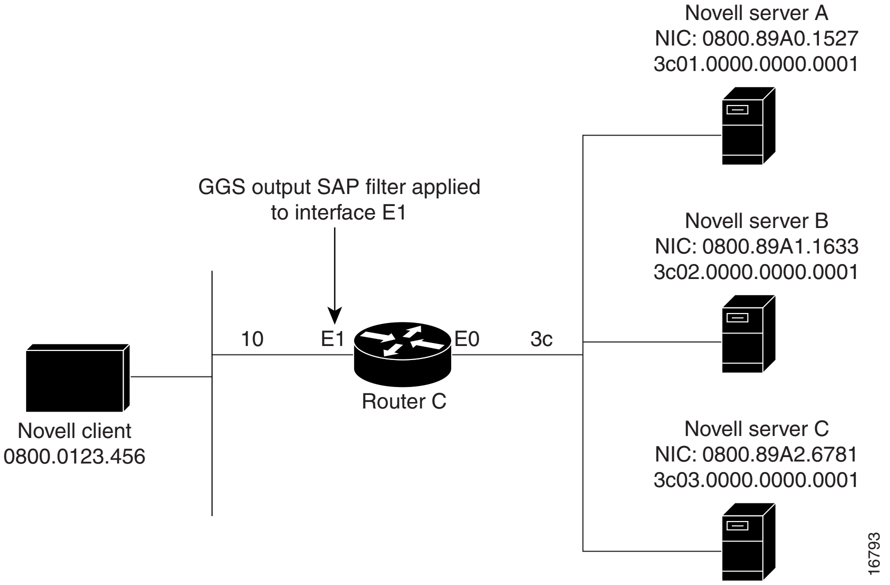
The following example shows how to configure Router C1. The first line denies server A. All other servers are permitted.

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

GGS SAP Response Filter Example

GGS SAP response filters as shown in Figure 6 allow a router to determine whether to forward information it receives about a service.

Figure 6 GGS SAP Response Filter



The following example shows how to configure GGS SAP response filters for Router C. When the client issues a GGS request, the output GGS filter denies a response from Novell Server A and permits responses from Novell servers B and C.

```
access-list 1000 deny 3c01.0000.0000.0001
access-list 1000 permit -1
interface ethernet 0
 ipx network 3c
interface ethernet 1
 ipx output-ggs-filter 1000
 ipx network 10
```

IPX NetBIOS Filter Examples

The following example shows how to use a NetBIOS host name to filter IPX NetBIOS frames. The example denies all outgoing IPX NetBIOS frames with a NetBIOS host name of Boston on Ethernet interface 0.

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

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

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

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

```
interface serial 1
 no ip address
 no keepalive
 ipx network 600
 ipx type-20-propagation
 ipx netbios input-access-filter bytes finigan
```

Helper Facilities to Control Broadcast Examples

The following sections show examples of how to control broadcast messages on IPX networks:

- Forwarding to an Address Example
- Forwarding to All Networks Example
- All-Nets Flooded Broadcast Example

Note that in the following examples, packet Type 2 is used. This type has been chosen arbitrarily; the actual type to use depends on the specific application.

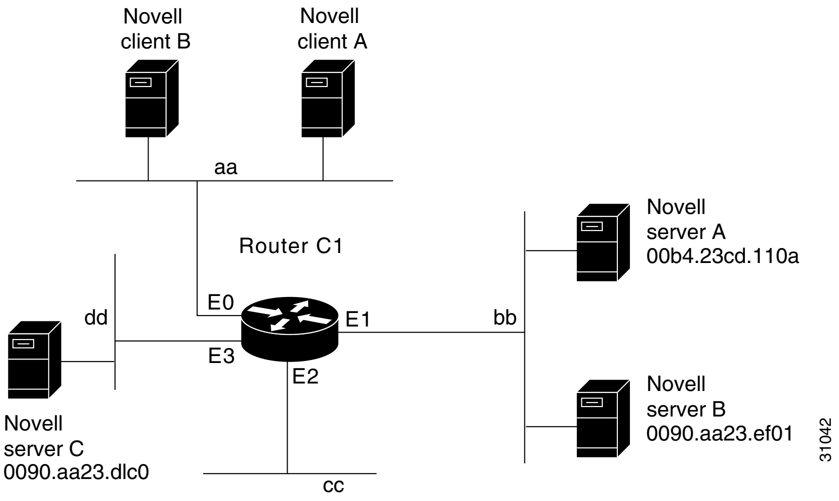
Forwarding to an Address Example

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

Figure 7 shows a router (C1) connected to several Ethernet interfaces. In this environment, all IPX clients are attached to segment aa, while all servers are attached to segments bb and dd. In controlling broadcasts, the following conditions are to be applied:

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

Figure 7 IPX Clients Requiring Server Access Through a Router



The following example shows how to configure the router shown in Figure 7. The first line permits broadcast traffic of Type 2 from network aa. The interface and network commands configure each specific interface. The **ipx helper-address** interface configuration commands permit broadcast forwarding from network aa to bb and from network aa to dd. The helper list allows Type 2 broadcasts to be forwarded. (Note that Type 2 broadcasts are chosen as an example only. The actual type to use depends on the specific application.) The **ipx type-20-propagation** interface configuration command is also required to allow Type 20 broadcasts. The IPX helper-list filter is applied to both the Type 2 packets forwarded by the helper-address mechanism and the Type 20 packets forwarded by Type 20 propagation.

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

This configuration means that any network that is downstream from network aa (for example, some arbitrary network aa1) will not be able to broadcast (Type 2) to network bb through Router C1 unless the routers partitioning networks aa and aa1 are configured to forward these broadcasts with a series of configuration entries analogous to the example provided for Figure 7. These entries must be applied to the input interface and be set to forward broadcasts between directly connected networks. In this way, such traffic can be passed along in a directed manner from network to network. A similar situation exists for Type 20 packets.

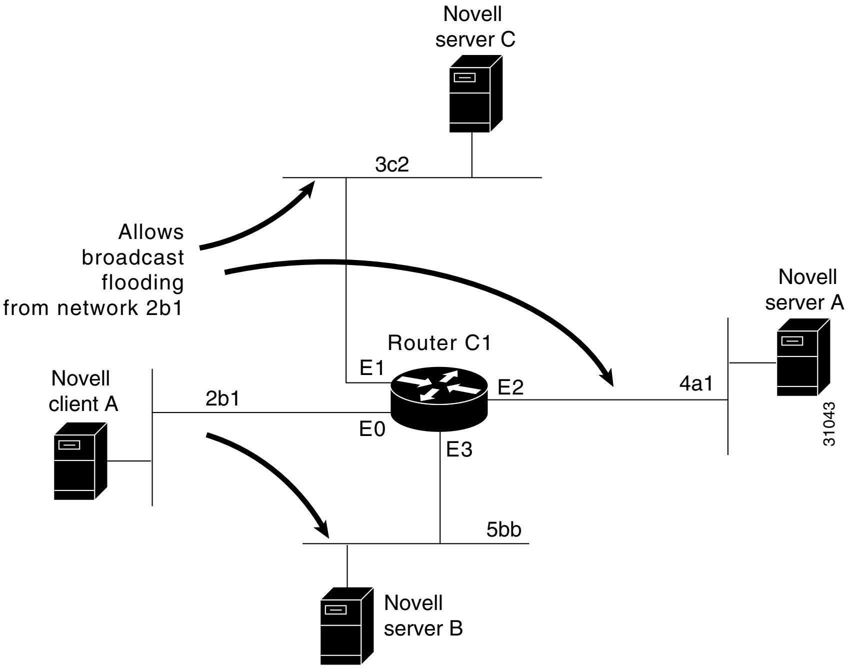
The following example shows how to rewrite the **ipx helper-address** interface configuration command line to direct broadcasts to server A:

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

Forwarding to All Networks Example

In some networks, it might be necessary to allow client nodes to broadcast to servers on multiple networks. If you configure your router to forward broadcasts to all attached networks, you are flooding the interfaces. In the environment illustrated in Figure 8, client nodes on network 2b1 must obtain services from IPX servers on networks 3c2, 4a1, and 5bb through Router C1. To support this requirement, use the flooding address (-1.ffff.ffff.ffff) in your **ipx helper-address** interface configuration command specifications.

Figure 8 *Type 2 Broadcast Flooding*



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

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

All-Nets Flooded Broadcast Example

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

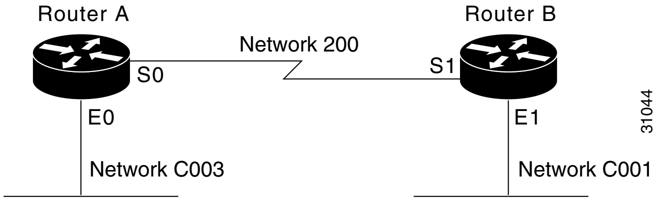
```
interface ethernet 0
 ipx network 23
 ipx helper-address -1.FFFF.FFFF.FFFF
```

IPX Accounting Example

The following example shows how to configure two Ethernet network segments that are connected via a serial link (see Figure 9). On Router A, IPX accounting is enabled on both the input and output interfaces (that is, on Ethernet interface 0 and serial interface 0), which means that statistics are gathered for traffic traveling in both directions (that is, out to the Ethernet network and out the serial link).

On Router B, IPX accounting is enabled only on the serial interface and not on the Ethernet interface, which means that statistics are gathered only for traffic that passes out the router on the serial link. Also, the accounting threshold is set to 1000, which means that IPX accounting will track all IPX traffic passing through the router up to 1000 source and destination pairs.

Figure 9 IPX Accounting Example



Configuration for Router A

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

Configuration for Router B

```
ipx routing
interface ethernet 1
 no ip address
 no keepalive
 ipx network C001
 no mop enabled
interface serial 1
 no ip address
 ipx network 200
 ipx accounting
 ipx accounting-threshold 1000
```

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.