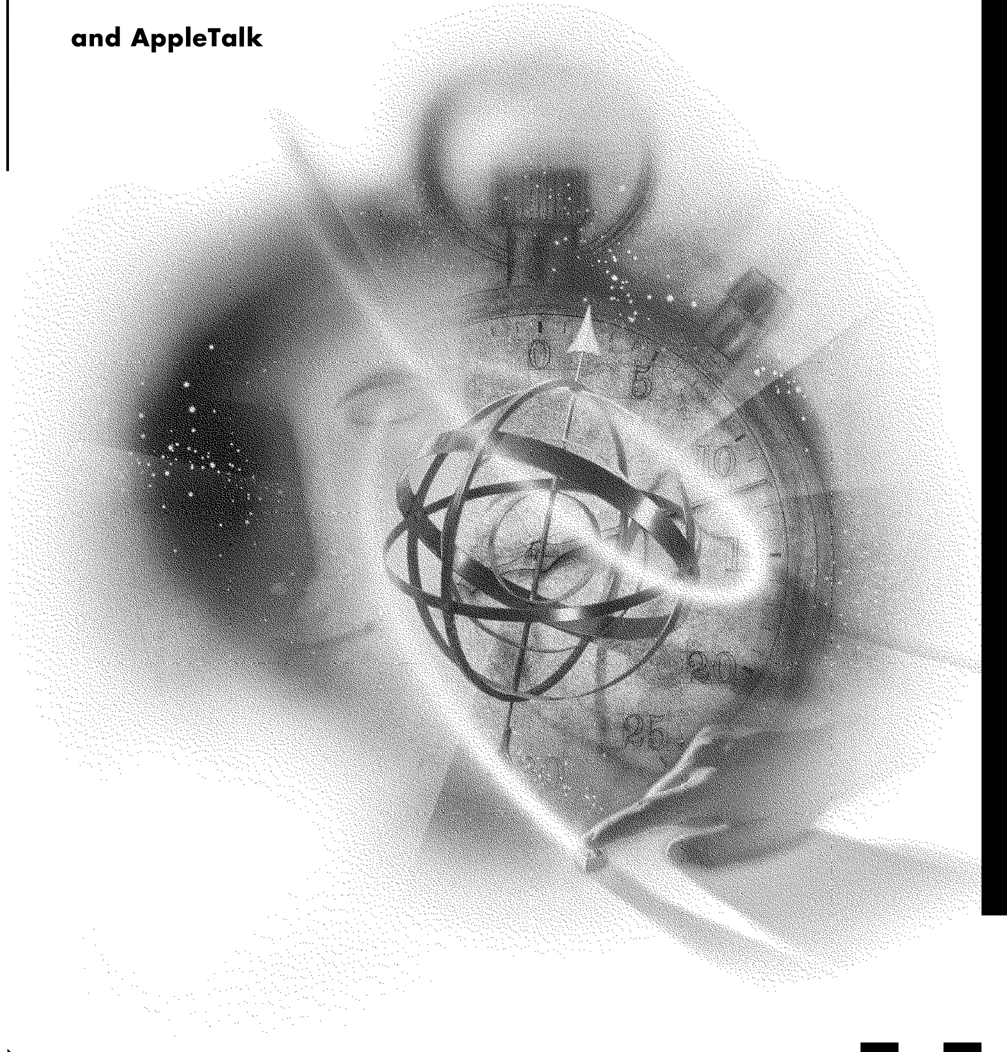
# IP Tunnels for IPX

# and AppleTalk



Connectivity Services

Novell®

## Legal Notices

Novell, Inc.
122 East 1700 South
Provo, UT 84606
U.S.A.

www.novell.com

IP Tunnels for IPX and AppleTalk
January 2000
104-001252-001

**Online Documentation:** To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

## Novell Trademarks

## Third-Party Trademarks

# Contents

# About This Guide

This guide provides the information you need to configure and manage the Novell Internet Access Server 4.1 IP tunnels for IPX and AppleTalk.

# 1 Setting Up

This section describes *IP tunneling* , the method by which two or more AppleTalk or Internetwork Packet Exchange™ (IPX™ ) networks exchange packets through an IP network.

## IP Tunneling for IPX

IPX uses the Open Data-Link Interface™ (ODI™ ) interface to pass packets through the IP tunnel. The IP tunnel sends each IPX packet across the TCP/IP network by encapsulating it in a User Datagram Protocol (UDP) packet. The tunnel driver at the destination router removes the UDP header from each incoming packet and passes it through ODI to IPX.

Encapsulating IPX packets in IP packets enables them to go through any TCP/IP supported media, such as Ethernet or token ring.

The TCP/IP network is the *medium* . The IP address is the *immediate address* , which performs the same function in the TCP/IP medium as the media access control (MAC) address performs in the Ethernet medium.

The Novell® Internet Access Server 4.1 routing software provides the following IP tunnel drivers:

- **IPRELAY** —WAN driver that models the IP internetwork as a collection of point-to-point permanent virtual circuits (PVCs) to tunnel IPX packets.
- **IPTUNNEL** —LAN driver that models the IP internetwork as a single IPX LAN to tunnel IPX packets.

To configure your router to use an IP tunnel, IP must be loaded and bound to the interfaces you plan to use. The IP tunnel requires local IP addressing information and can fail if IP is not bound to the network interface. For basic IP configuration procedures, refer to Setting Up in the *TCP/IP* documentation.

This topic contains the following sections:

## The IPRELAY Driver

The IPRELAY driver is a WAN driver that simulates a collection of point-to-point PVCs between routers. Each end point of each connection is an IP address. To establish a connection, only one side of the PVC must be configured. As long as one of the routers is aware of its peers, a connection can be made with those peers. A WAN call destination is created automatically for each IP peer.

## The IPTUNNEL Driver

You can also use the IPTUNNEL driver to enable IPX to use a TCP/IP network to communicate with other IPX nodes. You configure IPTUNNEL from the Novell Internet Access Server Configuration utility (NIASCFG) or from the command line.

The IPTUNNEL driver enables IPX to use a TCP/IP network to communicate with other IPX nodes. The IPTUNNEL driver models the IP internetwork as a single IPX LAN. To IPX, IPTUNNEL performs the same functions as a typical NetWare® LAN driver. The TCP/IP network operates as if it were a hardware network, passing packets among the IPX nodes connected to it.

IPTUNNEL is compatible with the Schneider & Koch SK-IPX/IP Gateway, which provides NetWare 2 compatibility. IPTUNNEL also serves workstations using either the Novell IP tunnel workstation driver, a component of the LAN WorkPlace® for DOS software, or the Schneider & Koch end node product for DOS.

When configuring the IP tunnel, you supply the IP addresses of other IPX routers that you plan to include in the tunnel. These other IPX routers are known as *peers* . Whenever IPX broadcasts a packet, the IP tunnel duplicates the packet and sends a copy to each peer.

To exchange routing and service information between IPX routers, IPX depends on broadcasting messages to every other NetWare server connected to the medium. However, because broadcast facilities are limited in TCP/IP networks, IPTUNNEL must handle broadcast traffic by duplicating the packet and sending a copy to each peer router.

IPTUNNEL presents a standard ODI driver interface to the NetWare system and handles IPX traffic like any other driver. You load the driver like any other, and then bind IPX to it to instruct IPX to receive and route packets over the TCP/IP network.

**IMPORTANT:** You should configure any connected group of peers so that all servers in the group have the IP addresses of all other servers in the group. Other configurations are possible but not recommended; they frequently create confusing—and often surprising—IPX routing topologies.

## Compatibility Between IPTUNNEL and IPRELAY

IPRELAY is compatible with IPTUNNEL. IPRELAY accepts packets from a LAN set up with IPTUNNEL as long as one of the routers on the LAN is running RIP.

Because IPRELAY allows point-to-point connectivity with the NetWare Link Services Protocol™ (NLSP™ ) software, NLSP operates more efficiently with IPRELAY than with IPTUNNEL. NLSP operates reliably over point-to-point connections; therefore, it has lower periodic traffic requirements.

Because IPRELAY works like any WAN driver, you can initiate and terminate IPRELAY connections from the Call Manager utility (CALLMGR). You can also use CALLMGR to reestablish lost IPRELAY connections. Additionally, CALLMGR identifies IPRELAY-to-IPTUNNEL connections with a (T) next to the remote peer IP address.

## How to Configure IPRELAY

Before you begin, you must make sure TCP/IP is enabled and bound to the interface.

To configure the IPRELAY tunnel, complete the following steps:

1. Load NIASCFG, then select the following parameter path:

   Select Configure NIAS > Protocols and Routing > Protocols > IPX

2. Set the Tunnel IPX Through IP parameter to Enabled .

3. Select Tunnel Configuration .

4. Add remote peers to the tunnel by selecting Remote Peers, then pressing Ins .

   The Insert New Remote Peer Address screen is displayed.

5. Type the remote peer IP address.

   This parameter adds an IP address to the peer list. If this parameter is not set, no peer is added. This is the most important parameter for a router that initiates connections.

6. Press Esc to return to the Tunnel Configuration menu.

7. If needed, configure the Transport Time parameter.

   If workstation connections fail because a server does not respond, increase this parameter. Select any value between 1 and 65535.

   **WARNING:** Do not change the User Datagram Protocol (UDP) port number. If the remote peer router is running IPTUNNEL, the local router automatically uses 213, an officially assigned UDP port number for IPX packets. If *both* routers are running IPRELAY, the local router automatically uses 2010. If you enter your own port number, the routers might not be able to communicate over the tunnel.

   The UDP Checksum should also not be changed from the default option (Enabled ). The UDP checksum improves data reliability.

8. Press Esc to return to the Internetworking Configuration menu; save your changes when prompted.

9. If you want these changes to take effect immediately, restart the router.

   If you want to configure other parameters, do so now, then restart the router when you are finished.

## How to Configure IPTUNNEL from NIASCFG

Before you configure IPTUNNEL, you must make sure that TCP/IP is enabled and bound to the interface.

To configure the IPTUNNEL driver, complete the following steps:

1. Load NIASCFG, then select the following parameter path:

   Select Configure NIAS > Protocols and Routing > Boards > Press Ins

2. Select IPTUNNEL from the list.

   The Board Configuration menu is displayed.

3. Configure the Board Name parameter.

4. Enter a valid address for the Peer IP Address parameter.

   If this parameter is not present, no peer is added. This is the most important parameter.

5. If needed, enter a value for the Local IP Address parameter.

   For IPX routing to work correctly, the IP tunnel must use a single local IP address consistently. The default value is the IP address of the first interface to which TCP/IP was bound.

6. Set the UDP Checksum parameter to Yes .

   Enabling this parameter improves data reliability.

7. If needed, enter a value for the UDP Port parameter.

   If you must communicate with nodes using products prior to Schneider & Koch SK-IPX/IP version 1.3, you can use port=59139. Otherwise, use the default value of 213, which is the officially assigned UDP port for IPX packets.

8. Press Esc .

   The new board appears at the end of the list on the Configured Boards screen.

9. Press Esc to return to the Internetworking Configuration menu.

10. Configure IPX and bind it to IPTUNNEL.

    For information about configuring various IPX functions, refer to Setting Up in the *IPX* documentation. Perform the procedures that apply to your situation.

11. Press Esc to return to the Internetworking Configuration menu; save your changes when prompted.

12. If you want these changes to take effect immediately, restart the router.

    If you want to configure other parameters, do so now, then restart the router when you are finished.

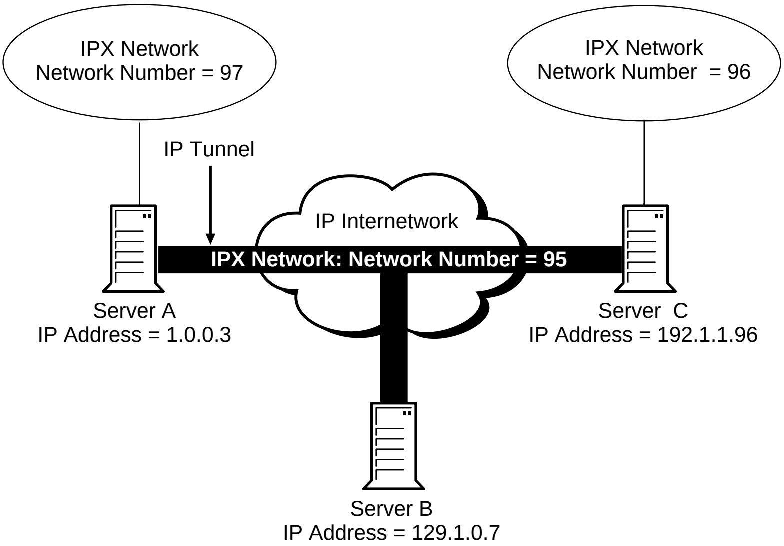
## How to Configure IPTUNNEL for Multiple Peers

Figure 1 shows how IPTUNNEL enables three NetWare servers—A, B, and C—to communicate over IPX network 95 as an IP tunnel through an IP internetwork.

**IMPORTANT:** IPTUNNEL duplicates and transmits every IPX broadcast packet in a UDP packet to each remote peer in a peer group. Because this can create a large amount of traffic on the network, you should have no more than 10 peers for any one node. We recommend that you use IPRELAY with NLSP for situations with more than 10 peers.

To configure IPTUNNEL for multiple peers, you must use LOAD and BIND commands from the command line as described below. The following commands are for the configuration example shown in Figure 1 .

**Figure 1 IPTUNNEL Configuration for Three Peers**



To configure IPTUNNEL on Server A, you enter the following commands at the Server A console:

```
LOAD IPTUNNEL PEER=129.1.0.7
LOAD IPTUNNEL PEER=192.1.1.96
BIND IPX to IPTUNNEL NET=95
```

The first two commands load IPTUNNEL and add entries on Server A for peer IP addresses 129.1.0.7 (Server B) and 192.1.1.96 (Server C). The third command binds IPX to IPTUNNEL.

To configure IPTUNNEL on Server B, you enter the following commands at the Server B console:

```
LOAD IPTUNNEL PEER=1.0.0.3
LOAD IPTUNNEL PEER=192.1.1.96
BIND IPX to IPTUNNEL NET=95
```

The preceding commands are almost identical to those in the Server A configuration. These commands add entries for peer IP addresses 1.0.0.3 (Server A) and 192.1.1.96 (Server C) and bind IPX to IPTUNNEL.

To configure IPTUNNEL on Server C, you enter the following commands at the Server C console:

```
LOAD IPTUNNEL PEER=1.0.0.3
LOAD IPTUNNEL PEER=129.1.0.7
BIND IPX TO IPTUNNEL NET=95
```

The preceding commands are almost identical to those in the Server B configuration. These commands add entries for peer IP addresses 1.0.0.3 (Server A) and 129.1.0.7 (Server B) and bind IPX to IPTUNNEL.

If needed, you can configure additional parameters with the LOAD IPTUNNEL command using the following format:

```
LOAD IPTUNNEL [PEER= remote IP address ]
[LOCAL= local IP address ] [CHKSUM={YES|NO}]
[PORT= UDP port number ] [SHOW={YES|NO}]
```

The NIASCFG parameters that are equivalent to the PEER , LOCAL , CHKSUM , and PORT parameters are explained in "How to Configure IPTUNNEL from NIASCFG."

The PEER parameter is equivalent to the Peer IP Address NIASCFG parameter.

The LOCAL parameter is equivalent to the Local IP Address NIASCFG parameter. If you configure IPTUNNEL from the command line, use the LOCAL parameter only with the first LOAD IPTUNNEL command.

The CHKSUM parameter is equivalent to the UDP Checksum NIASCFG parameter.

The PORT parameter is equivalent to the UDP Port NIASCFG parameter. If you configure IPTUNNEL from the command line, use the PORT parameter only with the first LOAD IPTUNNEL command.

**NOTE:** If you configure IPTUNNEL for multiple peer routers, use the LOCAL and PORT parameters only with the first LOAD IPTUNNEL command.

The SHOW parameter, available only with the command-based configuration, displays an IPTUNNEL configuration summary. If you load IPTUNNEL with SHOW set to YES (the default), the command displays the local IP address, the UDP port used, the peer list, and whether UDP checksums are enabled.

# IP Tunneling for AppleTalk

The AppleTalk Update-Based Routing Protocol (AURP) provides two features:

- Tunneling AppleTalk packets through an IP internetwork
- Exchanging routing information only when a change occurs in network topologies

If you need to connect two sites using a low bandwidth and costly WAN link, using AURP is the more economical way to provide AppleTalk connectivity between the two sites. Because less bandwidth is used to exchange routing information, more bandwidth can be used to carry user data.

To configure AppleTalk to use the IP tunnel, AppleTalk and TCP/IP must be enabled. For more information about configuring IP tunneling for AppleTalk, refer to Setting Up in the *TCP/IP* documentation and Setting Up in the *AppleTalk* documentation.

This topic contains the following sections:

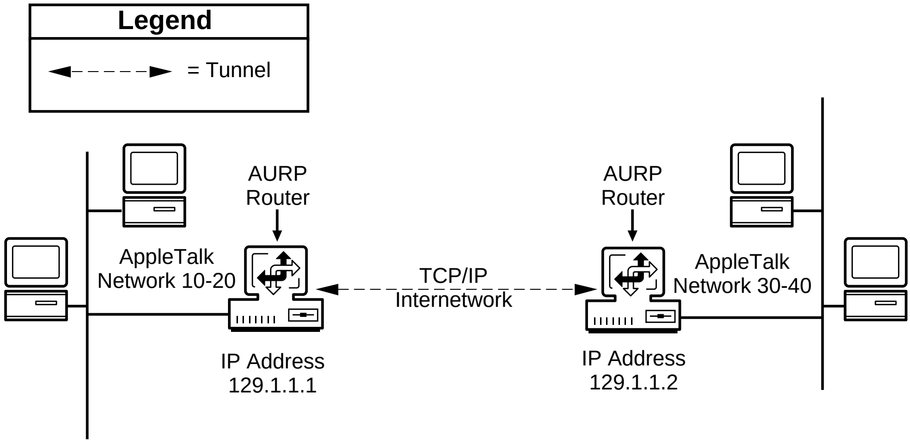
- Tunneling AppleTalk Packets
- How to Configure AURP

## Tunneling AppleTalk Packets

AppleTalk for NetWare uses AURP to encapsulate AppleTalk packets in IP packets. The forwarding AURP router encapsulates each AppleTalk packet in UDP and forwards it to the next AURP router (using UDP port 387 with checksums). The receiving AURP router removes the UDP and IP headers from the packet, then forwards it, like any other AppleTalk packet, to the destination AppleTalk network. For more information about AURP, refer to Understanding in the *AppleTalk* documentation .

Figure 2 shows two isolated AppleTalk networks connected by an IP tunnel.

**Figure 2 IP Tunnel Connecting Two AppleTalk Networks**



When configuring the IP tunnel for AURP, you supply the IP addresses of the AURP routers with which you plan to communicate. Generally, all AURP routers on an IP tunnel can communicate with one another. Each AURP router on a tunnel sends routing information about its local AppleTalk network to each of its peers on the tunnel. Because each AURP router is responsible for distributing its local network routing information, the receiving AURP routers on the tunnel do not need to forward the information to any of their AURP peer routers. This is similar to the operation of IPTUNNEL for IPX.

A *fully connected* tunnel is one in which all AURP routers on the tunnel are aware of and can communicate with one another. On a fully connected tunnel, the same number of routes should be reachable from each AURP router.

A *partially connected* tunnel is one in which not all AURP routers are aware of and can communicate with one another. A partially connected tunnel can provide network-level security. In a partially connected tunnel configuration, the routing tables on the different AURP routers can have different numbers of entries, and not all networks connected to these AURP routers are reachable by one another.

**IMPORTANT:** Partially connected tunnels can also be created accidentally if the router is not configured properly. For example, a network manager might create a partially connected tunnel accidentally by making an error when entering the list of peers with which the router should communicate.

## How to Configure AURP

When you enable AppleTalk, you can use the AppleTalk Configuration menu in NIASCFG to configure AURP to use the IP tunnel.

Before you begin, you must make sure TCP/IP is configured, enabled, and bound to at least one LAN or WAN interface before configuring AURP.

To configure AURP to use the IP tunnel, complete the following steps:

1. Load NIASCFG, then select the following path:

   Select Configure NIAS > Protocols and Routing > Protocols > AppleTalk

2. Enable the Tunnel AppleTalk Through IP (AURP) parameter.

3. Select AURP Configuration .

   The AURP Configuration menu is displayed.

   The UDP Port is always set to 387, and the UDP Checksum is always enabled.

4. Enter a value for the Local IP Address parameter.

   You select a unique address from a list of addresses with which other routers can establish connections.

5. Configure remote peers to which the router can tunnel AppleTalk packets.

   **5a** Select Remote Peers List and do one of the following:

   **If you are adding a new remote peer, press** Ins.

   **If you are modifying an existing remote peer, select the peer from the list.**

   **5b** Enter a valid address for the Remote IP Address parameter.

   **5c** Select Expert Options and configure the following parameters for each peer: Transmit Timeout , Maximum Transmit Retries , and Last Heard From Timeout Interval .

   These parameters apply only to the peer being configured.

6. Press Esc until you return to the AURP Configuration menu.

7. Select Expert Options .

8. Configure the Routing Update Interval parameter.

   This parameter applies to both configured and unconfigured peers.

**9** If you want the router to accept connections from any peers that were not configured in Step 5 , configure the following Expert Options :

**9a** Set the Connections From Unconfigured Peers parameter to Accept .

**9b** Configure the Last Heard From Timeout Interval parameter.

This parameter applies to all unconfigured peers.

**9c** Check the value of the Routing Update Interval parameter.

Verify that the value set for this parameter in Step 8 is acceptable for all unconfigured peers (and configured peers).

**10** Press Esc to return to the Internetworking Configuration menu; save your changes when prompted.

**11** If you want these changes to take effect immediately, restart the router or select Reinitialize System .

**IMPORTANT:** If you make changes to any of the parameters for AURP peers, reinitializing the system will cause all AURP peers connected to the router to disconnect and reconnect.

If you want to configure other parameters, do so now, then restart the router or reinitialize the system when you are finished.

**NOTE:** Do not bind AppleTalk to the interface you want to use as the IP tunnel. AURP uses the interface to which TCP/IP is bound as the IP tunnel. You can, however, bind AppleTalk to other interfaces.