Novell NetWare® 6

www.novell.com

TCP/IP
ADMINISTRATION GUIDE

Novell®

Legal Notices

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.

www.novell.com

NetWare TCP/IP Administration Guide
October 2001
103-000151-001

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

Novell Trademarks

ConsoleOne is a trademark of Novell, Inc.

Internetwork Packet Exchange and IPX are trademarks of Novell, Inc.

NetWare is a registered trademark of Novell, Inc., in the United States and other countries.

Novell Cluster Services is a trademark of Novell, Inc.

NetWare Loadable Module and NLM are trademarks of Novell, Inc.

Novell is a registered trademark of Novell, Inc., in the United States and other countries.

Novell Directory Services and NDS are registered trademarks of Novell, Inc., in the United States and other countries.

Third-Party Trademarks

All third-party trademarks are the property of their respective owners.

Contents

About This Guide

This guide provides the information you need to configure and manage the Novell® TCP/IP networking software. In addition to planning information, this guide provides troubleshooting tips, techniques, and tools, as well as the symptoms of and solutions to commonly occurring problems with the TCP/IP components.

Documentation Conventions

In this documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

Also, a trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

1 Understanding

This chapter introduces TCP/IP and provides an overview of the TCP/IP suite of protocols.

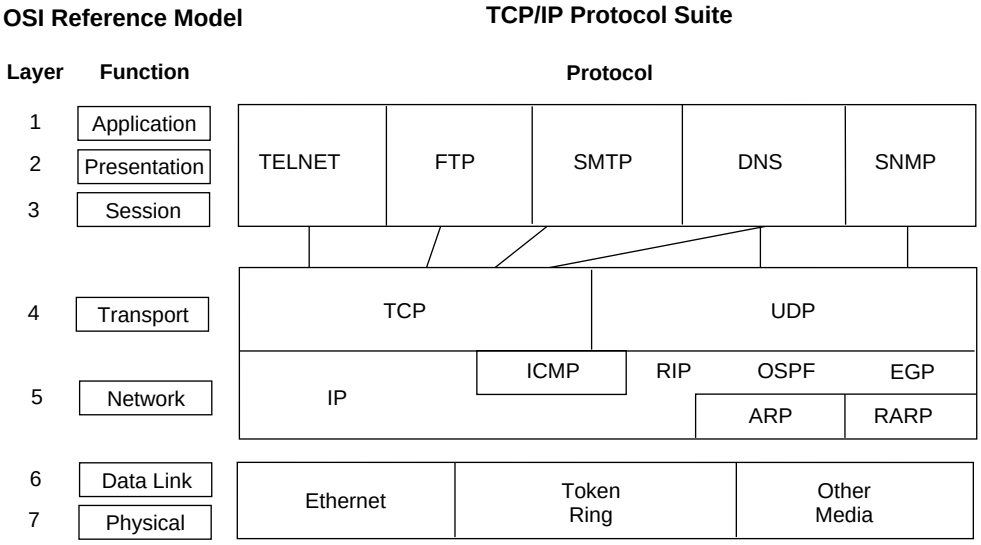
The following are discussed here:

The TCP/IP Suite of Protocols

The protocols in the TCP/IP suite roughly correspond to a network communications model defined by the International Organization for Standardization (ISO). This model is called the Open Systems Interconnection (OSI) reference model. The OSI model describes an ideal computer network system in which communication on the network occurs between processes at discrete and identifiable layers. Each layer on a given host provides services to the layers above it and receives services from the layers below it. Figure 1

illustrates the seven layers of the OSI reference model, as defined by ISO, and the roughly corresponding layers of the TCP/IP protocol suite.

Figure 1 OSI Reference Model



The layering system allows the developers to concentrate their efforts on the functions in a given layer. It is not necessary for designers to create all the mechanisms to send information across the network. They have to know only what services the software needs to provide to the layer above it, what services the layers below it can provide to the software, and which protocols in the suite provide those services.

Table 1 lists some of the more common protocols in the TCP/IP suite and the services they provide.

Table 1 TCP/IP Protocols

Protocol	Service
Internet Protocol (IP)	Provides packet delivery services (routing) between nodes.
Internet Control Message Protocol (ICMP)	Provides transmission of error and control messages between hosts and routers.

Protocol	Service
Address Resolution Protocol (ARP)	Maps IP addresses to physical addresses.
Transmission Control Protocol (TCP)	Provides reliable data-stream delivery service between end nodes.
User Datagram Protocol (UDP)	Provides unreliable datagram delivery service between end nodes.
File Transfer Protocol (FTP)	Provides application-level services for file transfer.
TELNET	Provides terminal emulation.
Routing Information Protocol (RIP)	Enables the exchange of distance vector routing information between routers.
Open Shortest Path First (OSPF)	Enables the exchange of link state routing information between routers.
Exterior Gateway Protocol (EGP)	Enables the exchange of routing information between exterior routers.

Overview of TCP/IP Protocol Usage

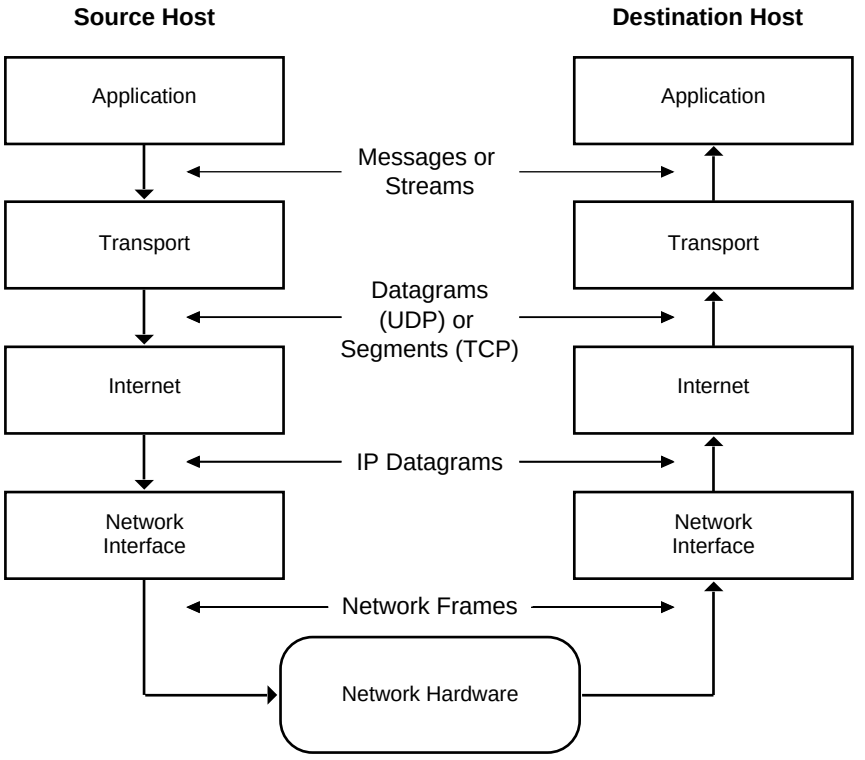
Applications developed for TCP/IP generally use several of the protocols in the suite. The layers of the protocol suite is also known as the *protocol stack*. User applications communicate with the top layer of the protocol suite. The top-level protocol layer on the source computer passes information to the lower layers of the stack, which in turn pass it to the physical network. The physical network transfers the information to the destination computer. The lower layers of the protocol stack on the destination computer pass the information to higher layers, which in turn pass it to the destination application.

Each protocol layer within the TCP/IP suite has various functions; these functions are independent of the other layers. Each layer, however, expects to receive specific services from the layer beneath it, and each layer provides specific services to the layer above it.

Figure 2 on page 14 shows the TCP/IP protocol layers. The layers at the same level on the source and destination computers are *peers*. For example, the application on the source computer and the application on the destination

computer are peers. Each layer of the protocol stack on the source computer communicates with its peer layer on the destination computer. From the perspective of the software developer or user, the transfer takes place as if the peer layers sent their packets directly to one another.

Figure 2 TCP/IP Model



An application for transferring files with TCP, for instance, performs the following operations to send the file contents:

1. The Application layer passes a stream of bytes to the Transport layer on the source computer.
2. The Transport layer divides the stream into TCP segments, adds a header with a sequence number for that segment, and passes the segment to the Internet (IP) layer. A checksum is computed over the TCP header and data.

3. The IP layer creates a packet with a data portion containing the TCP segment. The IP layer adds a packet header containing source and destination IP addresses.
4. The IP layer also determines the physical address of the destination computer or intermediate computer on the way to the destination host. It passes the packet and the physical address to the Data-Link layer. A checksum is computed on the IP header.
5. The Data-Link layer transmits the IP packet in the data portion of a data-link frame to the destination computer or an intermediate computer. If the packet is sent to an intermediate computer, steps 4 through 7 are repeated until the destination computer is reached.
6. At the destination computer, the Data-Link layer discards the data-link header and passes the IP packet to the IP layer.
7. The IP layer checks the IP packet header. If the checksum contained in the header does not match the checksum computed by the IP layer, it discards the packet.
8. If the checksums match, the IP layer passes the TCP segment to the TCP layer.
9. The TCP layer computes a checksum for the TCP header and data. If the computed checksum does not match the checksum transmitted in the header, the TCP layer discards the segment. If the checksum is correct and the segment is in the correct sequence, the TCP layer sends an acknowledgment to the source computer and passes the data to the application.
10. The application on the destination computer receives a stream of bytes, just as if it were directly connected to the application on the source computer.

Transport Layer Protocols

The Transport layer of the TCP/IP protocol suite consists of two protocols, UDP and TCP. UDP provides an unreliable connectionless delivery service to send and receive messages. TCP adds reliable byte stream-delivery services on top of the IP datagram delivery service.

The ports numbered between 1 and 1,023 are well-known port numbers. For dynamically bound ports, an application requests that UDP assign a port to

identify which port the process uses. The port must be in the range of 1,024 to 65,535.

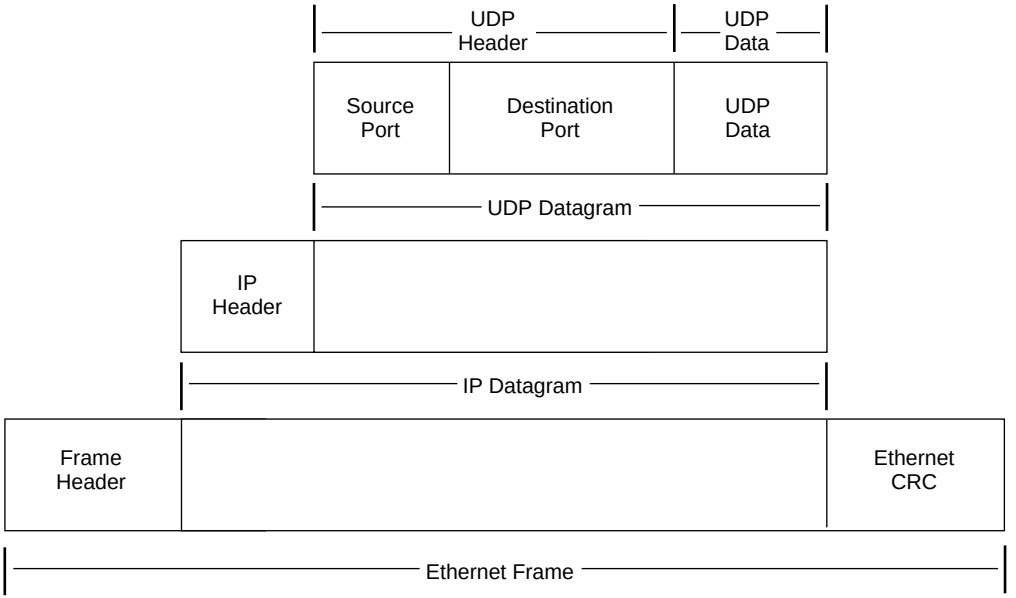
UDP

UDP identifies applications through *ports*. The protocol defines two types of protocol ports: well-known port assignments and dynamically bound ports. For well-known port assignments, certain UDP port numbers are reserved for particular applications. Then the application can direct UDP datagrams to that port.

UDP enables multiple clients to use the same port number and different IP addresses. The arriving UDP datagrams are delivered to the client that matches both the destination port number and address. (A socket consists of an IP address and the port number.) If there is no matching client or if the ICMP destination is unreachable then a port unreachable message is sent and the packet is dropped.

The UDP datagram is encapsulated in an IP datagram that, in turn, is encapsulated in physical frames. Figure 3 on page 17 shows a UDP datagram encapsulated in an IP datagram, which, in turn, is encapsulated in an Ethernet frame. It also illustrates how the concept of *layering*, discussed at the beginning of this section, affects the construction of packets sent across the network.

Figure 3 A UDP datagram encapsulated in an IP datagram which, in turn, is encapsulated in an Ethernet frame



In this example, the IP address transmits the IP datagram to the node. At that destination, the IP software extracts the UDP datagram and delivers it to the UDP-layer software. The UDP-layer software delivers the UDP data through the destination port to the receiving application. The process at that port uses the data in the UDP datagram. The UDP datagram also contains a source port to ensure that the destination process can reply correctly.

TCP

For applications that must send or receive large volumes of data, unreliable datagram delivery can become burdensome. Application programmers might have to develop extensive error handling and status information modules to track the progress and state of data transfer for every application. The TCP/IP suite of protocols avoids this problem by using TCP, a *reliable byte-stream delivery protocol*. TCP establishes a connection between two applications and sends a stream of bytes to a destination in exactly the same order that they left the source. Before transmission begins, the applications at both ends of transmission obtain a TCP *port* from their respective operating systems. These are analogous to the ports used by UDP. The application initiating the transfer, known as the client side, generally obtains a port dynamically. The application

responding to the transfer request, known as the server side, generally uses a well-known TCP port. The client side is typically the active side and initiates the connection to the passive server side.

Like the UDP datagrams, TCP *segments* are encapsulated in an IP datagram. TCP *buffers* the stream by waiting for enough data to fill a large datagram before sending the datagram. The stream is *unstructured*, which means that before transmission of data, both the sending and receiving applications must agree on the meaning of the contents of the stream. The TCP protocol uses *full-duplex* transmission. Full duplex means that two data streams can flow in opposite directions simultaneously. Thus, the receiving application can send data or control information back to the sending application while the sending application continues to send data.

The TCP protocol gives each segment a sequence number. At the receiving end of the connection, TCP checks successive sequence numbers to ensure that all the segments are received and processed in the order of the sequence numbers. The receiving end sends an acknowledgment to the sender for the segments received. TCP enables the sender to have several outstanding segments before the receiver must return an acknowledgment. If the sending node does not receive an acknowledgment for a segment within a certain time, it retransmits that segment. This scheme, called *positive acknowledgment with retransmission*, ensures that the stream delivery is reliable.

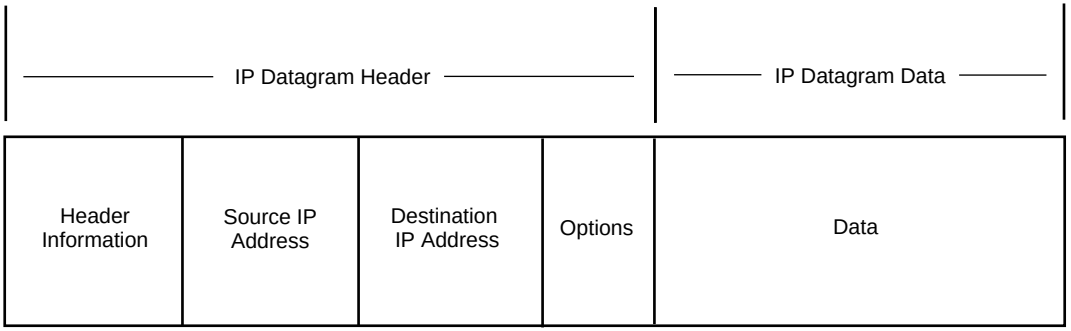
Internet Protocol

In the TCP/IP protocol suite, all packets are delivered by the IP *datagram delivery* service. Packet delivery is not guaranteed by this service. A packet can be misdirected, duplicated, or lost on the way to its destination. The service is *connectionless* because all packets are transmitted independently of any other packets. This is in contrast to a telephone network, for instance, where a connection is established and maintained.

To keep track of the delivery status, TCP/IP applications using the IP datagram delivery service expect to receive replies from the destination node.

IP defines the form that packets must take and the ways that packets are handled when they are transmitted or received. The form the packet takes is called an *IP datagram*. It is the basic unit of information that is passed across a TCP/IP network. The IP datagram consists of a header and a data section. The header section contains the sender's (source) IP address and the receiver's (destination) IP address and other information. Figure 4 on page 19 shows the general form of an IP datagram.

Figure 4 Basic frame of an IP Datagram



The IP address for a node is a logical address and is independent of any particular hardware or network topology. It has the same form, regardless of the media type. The IP address (version 4) is a 4-byte (32-bit) numeric value that identifies both a network and a local host or node (computer or other device) on that network. The 4-byte IP address is usually represented in dotted decimal notation. Each byte is represented by a decimal number, and periods separate the bytes, for example, 129.47.6.17.

The Data-Link layer transmits IP packets in the data section of its physical frame. Because IP supports a 64-KB packet length, an IP datagram might not fit in a data-link frame. Also, in traveling to its destination, a datagram can traverse many different media with different physical frame lengths. An IP router might have to forward a packet across media in which the inbound and outbound frame lengths differ.

To handle these potential problems with packet transmission, IP specifies a method for breaking datagrams into *fragments*. The fragments are *reassembled* when they arrive at the final destination. Reassembling fragments reconstructs the entire IP datagram.

Path Maximum Transfer Unit

The maximum transfer unit (MTU) is the largest amount of data that can be transferred across a given physical network. For local area networks, such as Ethernet, the MTU is determined by the network hardware. For wide area networks that use serial lines to interconnect packet switches, the MTU is determined by software.

The Path MTU is the smallest of all MTUs, for the hops along a path from the source host to the destination host. The Path MTU governs the size of the

largest IP packet that can be sent across the path without fragmentation. This feature conforms to RFC 1191.

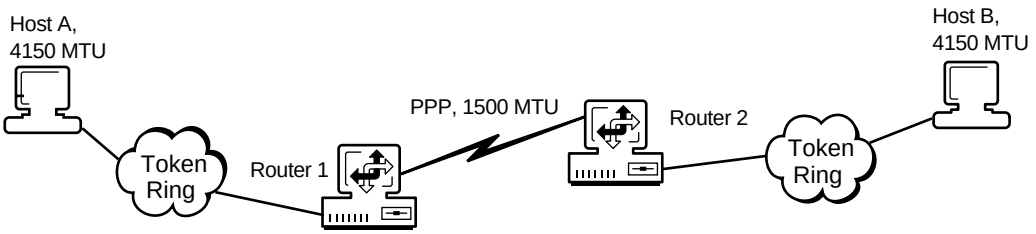
This feature is automatically enabled when you enable TCP/IP.

Path MTU Discovery Process

There are two advantages to this feature. The Path MTU avoids fragmentation anywhere along the path and it reduces the protocol overhead.

The Path MTU discovery process prevents fragmentation between two routers. Figure 5 illustrates a sample Path MTU discovery process, followed by an example of the steps involved.

Figure 5 A sample Path MTU discovery process



The following describes the steps involved in the sample Path MTU discovery process illustrated above:

1. Host A opens a File Transfer Protocol (FTP) connection to Host B.
2. Host A and Host B negotiate the maximum segment size (MSS) during their connection. This is the largest TCP segment that a host can send across a network. The MSS in Figure 5 is 4,110 bytes, which is 4,150 bytes minus 40 bytes for the IP and TCP headers.
3. Host A sends a 4,150-byte packet to 4,110 bytes of data and 40 bytes of header information to Host B. The Don't Fragment (DF) flag in the IP header is set to yes in Host A.
4. Router 1 receives the packet from Host A. Then Router 1 determines that the packet is larger than 1,500 bytes, which is the maximum packet size that can be sent over a PPP network.
5. Router 1 sends Host A an ICMP destination unreachable error message. This message indicates that Router 1 must fragment packets larger than 1,500 bytes.

6. Host A receives the error message from Router 1. In response, it adjusts the maximum segment size to 1,460 bytes.
7. Host A resends the data from Step 3. Each packet consists of 1,460 bytes of data and 40 bytes of header information.
8. Router 1 accepts the packets and forwards them to Router 2, which then sends them on to Host B.

Assigning IP Network Addresses

IP network addresses should be assigned by one person at your company. We recommend that a network administrator assign IP network addresses. Therefore, to obtain a new address, see your network administrator or if you are a network administrator this section would help you assign IP network addresses.

For a node using the TCP/IP protocol suite to communicate with other nodes, including nodes on other private networks and on the Internet, an IP network address is required. Your IP network address could be determined in one of the following ways:

- If you are accessing the Internet through an Internet Service Provider (ISP), you can be assigned an IP address by your ISP.
- If you are connected directly to the Internet community or if you cannot connect to the Internet using the registered IP address range you were assigned by your ISP, contact the following organization:

  Network Solutions, Inc.
  Attn.: InterNIC Registration Services
  505 Huntmar Park Dr.
  Herndon, VA, USA 20170

  E-mail: hostmaster@internic.net

  Web address: http://nic.ddn.mil or http://192.112.36.5
- If your network is not attached to the public Internet community, you can select an arbitrary IP network number. However, if you plan to attach your network to the Internet later, you should use the guidelines in RFC 1918.

The addresses for all the nodes on the network must meet the following criteria:

- All addresses within a network must use the same prefix. For example, any node on network 129.47 must have an address in the form 129.47.x.x.
- Each node must have a unique IP address.

Historic IP Address Classes

Each 4-byte IP address is divided into two parts:

- A *network* portion, which identifies the network
- A *host* portion, which identifies the node

IP addresses are differentiated into three classes, based on the two most significant bits of the first byte. This is done so that routers can efficiently extract the network portion of the address.

This division can occur at any one of three locations within the 32-bit address. These divisions correspond to the three IP address classes: Class A, Class B, and Class C. Regardless of address class, all nodes on any single network share the same network portion; each node has a unique host portion.

Class A Addresses

A Class A IP address consists of a 1-byte network portion followed by a 3-byte host portion, as shown in Figure 6 on page 23. The highest-order bit of the network portion is always set to 0. Thus, within an internetwork, there can be a total of 126 Class A networks (1 through 126), with more than 16 million nodes in each (networks 0 and 127 are reserved).

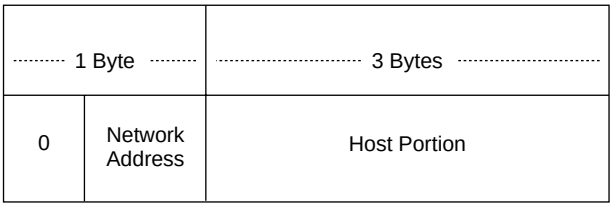
The format of a Class A address is as follows:

```
0nnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh
```

where *n* represents the network address and *h* represents the host address.

Class A addresses contain 7 bits of network address and 24 bits of host address.

Figure 6 A Class A IP address



Class B Addresses

A Class B IP address consists of a 2-byte network portion followed by a 2-byte host portion, as shown in Figure 7. The two highest-order bits of the network portion are always set to 10. Thus, within a single internetwork there can be approximately 16,000 Class B networks (128.0 through 191.255), with more than 65,000 nodes in each.

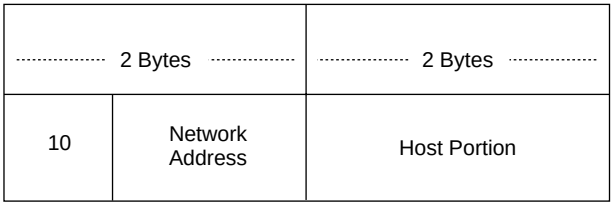
The format of a Class B address is as follows:

`10nnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh`

where *n* represents the network address and *h* represents the host address.

Class B addresses contain 14 bits of network address and 16 bits of host address.

Figure 7 A Class B IP address



Class C Addresses

A Class C IP address consists of a 3-byte network portion followed by a 1-byte host portion, as shown in Figure 8 on page 24. The three highest-order bits of the network portion are always set to 110. Within a single internetwork, there can be approximately 2 million Class C networks (192.0.0 through 223.255.255), with up to 254 nodes in each.

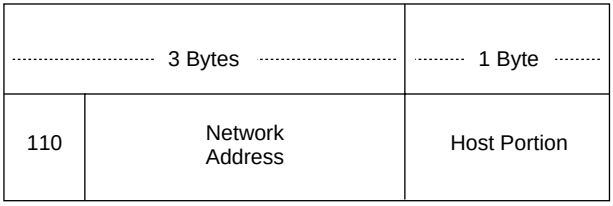
The format of a Class C address is as follows:

`110nnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh`

where *n* represents the network address and *h* represents the host address.

Class C addresses contain 21 bits of network address and 8 bits of host address.

Figure 8 A Class C IP address



Identifying Network Classes

The first byte of an IP address identifies which of the three network classes that address belongs to. The ranges for that first byte are as follows:

- Class A: 1 to 126 (1.h.h.h to 126.h.h.h)
- Class B: 128 to 191 (128.n.h.h to 191.n.h.h)
- Class C: 192 to 223 (192.n.n.h to 223.n.n.h)

An IP address beginning with 154 is a Class B address. The first two bytes of the address are represented by *n* for the network portion of the address, and the last two bytes are represented by *h* for the host portion. For example, an IP address of 154.1.0.3 means the IP network portion is 154.1.0.0 and the host portion on that network is #.#.0.3.

The network portion of an IP address should be the same for all nodes on that network. Each node connected to the network must have a unique IP host address assigned to it.

HINT: The key to selecting a number for the host portion of the IP address is to ensure that the number selected is unique, that is, that no other host on the network has the same IP address.

Selecting an Appropriate Address Class

When selecting an IP address class, you must decide on both network numbers and host address portions. Because the first 1, 2, or 3 bits of the IP address determine how the entire address is to be interpreted and where the division between the network address and host address portion is to occur, you should know the consequences of your choice. When deciding on a network class, you should consider the number of IP nodes to be supported on your network and the number of networks you plan to configure.

For example, if you use Class C addresses (the first 3 bits of the IP address are 110 binary), then you are restricted to 254 nodes. However, the number of nodes available can be altered by using subnets. Before selecting an IP address class, see "Creating Subnets" on page 28.

Reserved IP Addresses

The IP addressing rules reserve the following types of IP addresses for special purposes:

- **Network addresses**—IP addresses in which the host portion is set to all zeros. For example,129.47.0.0 is the network address (or network number) for a Class B network. Network addresses identify networks rather than nodes on a network. By convention, no node is ever assigned a host portion consisting of all zeros.
- **Broadcast addresses**—Addresses in which the host portion is set to all ones. A packet with a broadcast address is destined for every node on the network. By convention, no node is ever assigned a host portion consisting of all ones.
- **Loopback addresses**—Addresses that cause the protocol software to return data without sending traffic across a network. Network address 127.0.0.0 and all host addresses on that network (for example, 127.0.0.1) are reserved.
- **Multicast addresses**—Addresses that are used to send packets to a group of hosts or routers. They range from 224.0.0.1 to 239.255.255.255.
- **Reserved addresses**—Addresses in which the network portion consists of all zeros or all ones.

Physical and IP Addresses

Each node has a *physical address* for the specific hardware device that connects it to a network. For instance, a physical address on an Ethernet network is a 6-byte numeric value, such as 08-00-14-57-69-69. It is assigned by the manufacturer of the Ethernet interface hardware. X.25 networks, which conform to the specification of the ITU-T (International Telecommunications Union, Telecommunications sector), previously CCITT, use the X.121 standard for physical addresses, which consist of 14-digit numbers.

NOTE: Physical addresses are also called media access control (MAC) addresses. Throughout the rest of this section, all references to MAC or physical addresses assume physical addresses on Ethernet, token ring, or FDDI networks.

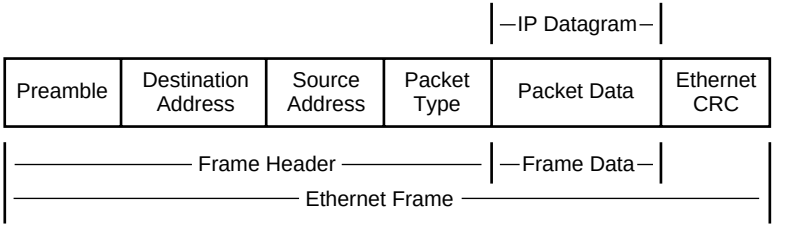
Since IP uses a 32-bit address and Ethernet uses a 48-bit Ethernet address there is a conflict. To associate the IP address to a physical address on an Ethernet network, a mapping must occur between the two types. The address resolution protocol (ARP) provides a mapping between the two different forms of addresses.

IP Address to Physical Address Translation

Each physical medium has its own *physical* address for nodes on that medium. The physical addresses are also called MAC addresses. Ethernet and token ring networks use 6-byte MAC addresses. ARCnet uses a 1-byte MAC address.

IP addresses are independent of the hardware. When an IP packet is transmitted on the network, it is first *encapsulated* within the physical frame used by that network. Figure 9 shows an IP packet encapsulated in an Ethernet frame. The IP packet contains an Internet address for a node, but the Ethernet frame must have a physical address for it to be delivered on the data-link network. Therefore, the sending node must be able to map an IP address to a physical hardware address.

Figure 9 A packet encapsulated in an Ethernet frame



Mapping Internet Addresses to Physical Addresses

When an IP address is mapped to a physical, or MAC, address, ARP is used on broadcast networks such as Ethernet, token ring, and ARCnet. When a node uses IP to send a packet, it must determine which physical address on the network corresponds to the destination IP address. To find the physical address, the node broadcasts an ARP packet containing the destination IP address. The node with the specified destination IP address sends its physical address back to the requesting node.

Address Resolution Cache

To speed packet transmissions and reduce the number of broadcast requests that must be examined by every node on the network, each node keeps an *address resolution cache*. Each time the node broadcasts an ARP request and receives a response, it creates an entry in its address resolution cache. The entry maps the IP address to the physical address.

When the node sends an IP packet, it looks up the IP address in its cache and uses the physical address, if found. The node broadcasts an ARP request only if the IP address is not in its cache.

ARP Cache Update Timeout

ARP Cache Update Timeout is a configurable parameter used to specify the timeout period for an entry to be removed from the ARP table, if the entry has not been updated. The ARP Cache Update Timeout value should be greater than or equal to the ARP Cache Stale Timeout value.

ARP Cache Stale Timeout

ARP Cache Stale Timeout ia a configurable parameter used to specify the timeout period for an entry to be removed from the ARP table, if the entry has not been used for some time. The ARP Cache Stale Timeout value should be less than or equal to the ARP Cache Update Timeout value.

Creating Subnets

One IP network can be divided into smaller networks, called subnets. The following are reasons to divide your network:

- **Use multiple media**—It can be impossible, inconvenient, or too expensive to connect all nodes to a single network medium when these nodes are too far apart or already connected to different media.
- **Reduce congestion**—Traffic between nodes on a single network uses network bandwidth. As a result, more bandwidth is required when you have more nodes. Splitting nodes into separate networks reduces the number of nodes on a data-link network. Fewer nodes generate less traffic and, as a consequence, less congestion.
- **Reduce CPU use**—Reducing CPU use on connected nodes is similar to reducing congestion. More nodes on a network cause more broadcasts on that network. Even if a broadcast is not sent to a particular node, each node on a network must react to every broadcast before deciding to accept it or discard it.
- **Isolate a network**—By splitting a large network into small networks, you limit the impact of one network's problems on another. Such problems can include network hardware failures, such as an open Ethernet tap, or software failures, such as a broadcast storm.
- **Improve security**—On a broadcast network medium such as Ethernet, each node on a network has access to all packets sent on that network. By enabling sensitive network traffic on only one network, other network monitors can be prevented from accessing this sensitive traffic.
- **Make efficient use of IP address space**—If you are using a Class A or B network number and have multiple small physical networks, you can divide the IP address space into multiple IP subnets and assign them to individual physical networks. Another option is to obtain several Class C network numbers, although this is less desirable.

  For more information about creating subnets, see the following:

  - “Subnet Addresses and Masks” on page 29
  - “Subnet Zero” on page 32
  - “Variable Size Subnets” on page 32
  - “Assigning Subnet Addresses” on page 33
  - “Broadcast Addresses” on page 34
  - “Multicast Addresses” on page 34

Subnet Addresses and Masks

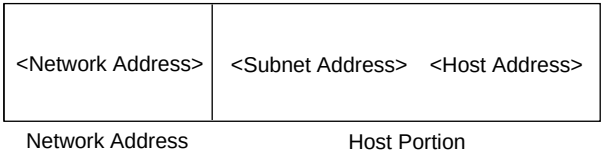
Communication between a node on a local subnet and a node on a different subnet is similar to communication between nodes on two different networks. To a user, routing between subnets is transparent. Internally, the IP software recognizes any IP addresses that are destined for a remote subnet and sends those packets to the router on that subnet.

As in network-to-network communication, the routing information for communication between subnets is maintained in the routing table (by IP).

When a network is divided into subnets, the host address portion of the IP address is divided into two parts, just as the IP address itself is divided into two parts. The host address portion specifies both the subnet of the IP network and the node on that subnet.

The 4-byte IP address consists of a network address and a host portion, as shown in Figure 10.

Figure 10 A 4-byte IP address

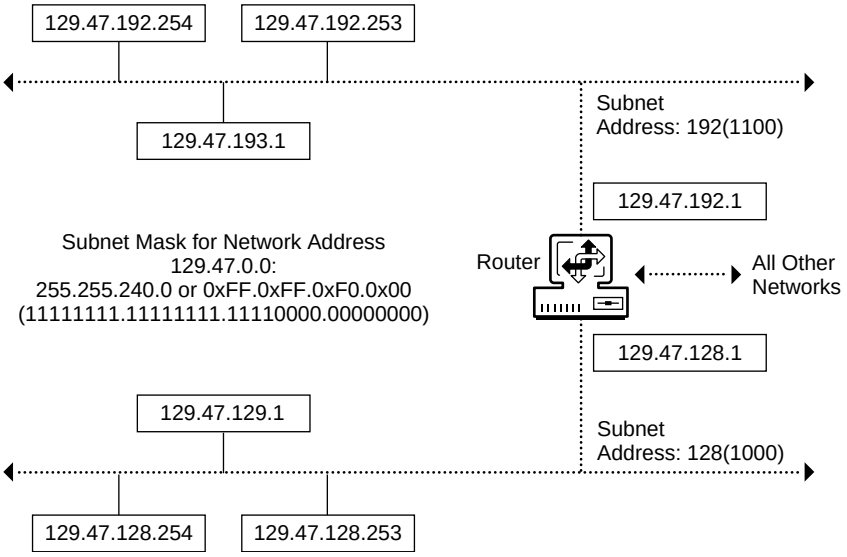


For instance, if a network has the Class B IP network address portion 129.47, the remainder of the IP address can be divided into subnet addresses and host addresses. Controlled by the local network administrator, this division allows the most flexibility for network development at the local site. For example, the subnet address could comprise 4 bits of the remaining 2 bytes. This allows 15 subnets, each with 4,094 nodes. Or, in another case, the subnet address could comprise 8 bits, allowing 255 subnets (a subnet address of all ones is not valid), each with 254 nodes.

NOTE: NetWare routing software supports the use of all zeros in the subnet field (subnet zero). However, a subnet field with all ones denotes all subnets of a particular network; therefore, a subnet field with all ones cannot be used as a local IP address.

Figure 11 on page 30 shows a single IP network divided into two subnets. The router shown has physical attachments and IP addresses on both subnets (129.47.128.1 and 129.47.192.1). It might also have physical devices and IP addresses (*nn.nn.nn.nn*) connecting it to other networks.

Figure 11 A single IP network divided into two subnets



A *subnet mask* indicates how the host portion of the IP address is divided into a subnet address and a local host portion. The network mask is a 32-bit number with all ones for all network and subnet address portions, and all zeros for the host field. With a Class B network portion of 129.47 and a 4-bit subnet address, for instance, the subnet mask consists of 20 ones and 12 zeros. In essence, a subnet mask locally extends the network address portion of an IP address and reduces the host portion.

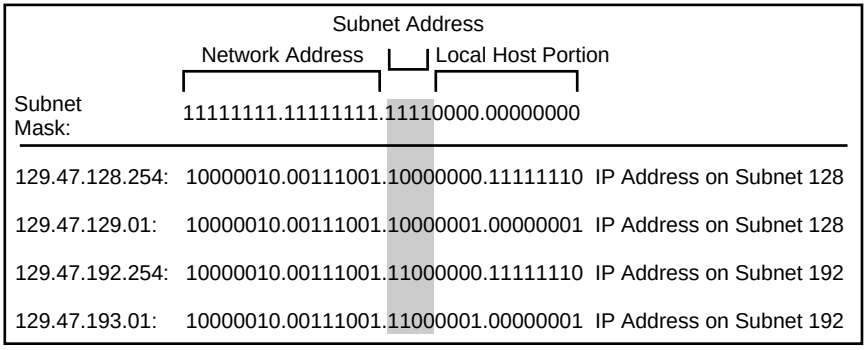
Table 2 on page 31 shows an example of a Class C subnet with an IP address of 200.2.1.209. To create a subnet address, bits are taken from the local host portion. As the size of the subnet mask increases, the number of hosts decreases and the number of subnets increases.

Table 2 Subnet Masks with Class C Addresses

Class C IP Address 200.2.1.209	Network Number	Subnet Number	Host Number	Available Networks, Subnets, and Hosts
FF.FF.FF.0	200.2.1.0	None	0.0.0.209	1 network, 0 subnets, and 254 hosts
FF.FF.FF.E0	200.2.1.0	200.2.1.192	0.0.0.17	7 subnets and 30 hosts per subnet
FF.FF.FF.F0	200.2.1.0	200.2.1.208	0.0.0.1	15 subnets and 14 hosts per subnet

Figure 12 shows examples of IP network addresses, their relationship to the subnet mask, and the corresponding subnets.

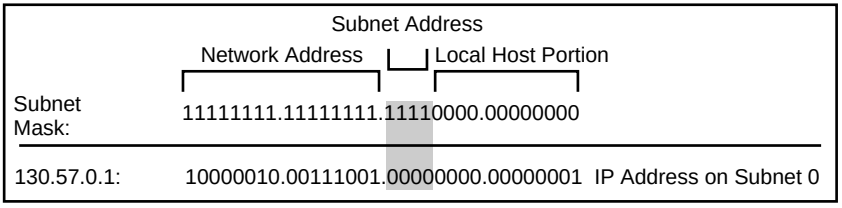
Figure 12 Examples of IP network addresses



Subnet Zero

Subnet zero is a subnet with all the bits in the subnet field of the IP address set to 0. For example, subnet 130.57.0.0, with a mask of 255.255.240.0, is a subnet zero of network 130.57, as shown in Figure 13 on page 32.

Figure 13 A subnet zero



The official IP specification reserves the subnet addresses with all zeros and all ones and does not allow them to be used as subnet addresses. However, this policy wastes one subnet in the IP address space. To counteract this limitation, Novell's TCP/IP implementation enables the use of subnet zero.

Variable Size Subnets

The subnets of a network can have different length subnet masks, called *variable length subnet masks*. These subnets are called variable because the size, or length, of the subnet varies from subnet to subnet.

A subnet mask defines the number of bits that can be used to define the subnet and the number of bits to define the host. As the subnet mask increases, the number of hosts on a subnet decreases. As the subnet mask decreases, the number of hosts that can be defined increases.

Some network configurations have individual subnets with a large number of hosts and other subnets with a small number of hosts. Using the same subnet masks on all subnets can mean either of the following:

- The mask is too small and you do not have enough subnet numbers for all your subnets.
- The mask is too big and you do not have enough host IDs for all your hosts on a subnet.

If the mask is too small or too big, use a variable size subnet. By varying the size of the subnet mask used on a network, you can match the number and size of subnets to your configuration.

For example, subnet 16 of network 130.57.0.0 with mask 255.255.240.0, 130.57.16.0, can be further divided into 16 sub-subnets with 256 hosts each. (Actually, this division creates 15 sub-subnets with 254 hosts each because sub-subnet 130.57.31.0, host 0, and host 255 are not used.)

NOTE: OSPF and RIP II recognize subnet masks and support variable size subnets. RIP I does not work when the network is partitioned into variable length subnets because RIP I assumes that all subnets belonging to the same network use the same subnet mask.

Assigning Subnet Addresses

HINT: Because RIP I packets do not carry subnet mask information, the RIP I routing protocol imposes several restrictions on the use of subnets. If you are using RIP I, use the same subnet mask for all subnets belonging to the same network. Using RIP II lifts this restriction.

If you are installing the routing software on a network with subnets, use the subnet mask already established for the network.

Subnet addresses and host addresses are typically assigned in numeric order, where both the subnet and host addresses are assigned from the right edge of their field. By this method, the border between the subnet address and the host address becomes fixed when the first subnet (subnet address = 1) is assigned. If the number of hosts on a subnet or the number of subnets required exceeds the limits of the subnet mask, using this method makes it difficult to adjust the subnet mask because each host must be renumbered.

To prepare for changes in the size of the subnet mask, RFC 1219 suggests that subnets be assigned from the *left* of the subnet address field, and that hosts be assigned, in numeric order, from the *right* of the host address field. In this way, the subnet bits become a *mirror image* of the host bits. (You must still select an initial subnet mask and use it for all subnets in the network.) For example, to apply this method to a Class B IP network with a subnet mask of 255.255.255.0, you assign subnet addresses as follows:

1000 0000 (Decimal 128)
0100 0000 (Decimal 64)
1100 0000 (Decimal 192)
0010 0000 (Decimal 32)
...
0000 0001 (Decimal 1)
0000 0010 (Decimal 2)
0000 0011 (Decimal 3)
0000 0100 (Decimal 4)

Then, you assign host addresses on each subnet as follows:

. . .

Using this method leaves a buffer zone between the subnet and host addresses, which enables future network growth.

The method of assigning subnet addresses described in this section summarizes the method suggested in RFC 1219, *On the Assignment of Subnetwork Numbers*. For a complete description of this method, refer to RFC 1219.

Broadcast Addresses

There are four types of broadcast addresses: directed broadcasts, subnet directed broadcasts, all-subnets directed broadcasts, and limited broadcasts. A directed broadcast has a destination IP address with the network portion of the IP address set to Class A, B, or C network, and the host field set to all ones. Directed broadcasts are sent to all hosts on the specified network.

If the network is divided into subnets, each subnet has a subnet directed broadcast. A subnet directed broadcast has an IP address with the network field set to the network identifier, the subnet field set to the subnet identifier, and the host field set to all ones.

An IP address with both the subnet and host field set to all ones is interpreted as a broadcast directed to all the subnets on the network. That is, the first router on the specified network broadcasts the IP address to one of its subnets. If broadcast forwarding is enabled, the receiving routers in that network forward the broadcast to other subnets.

An IP address with all bits set to 1, that is 255.255.255.255, is called a limited address. It is directed to all hosts on the subnet from which the broadcast originated.

Multicast Addresses

A multicast address is used to send packets to a group of hosts or routers. A packet with a multicast address is received by all hosts and routers belonging to that multicast group. Class D addresses are reserved for multicast addresses. They range from 224.0.0.1 to 239.255.255.255.

Novell's TCP/IP implementation uses five multicast addresses. Two are used by OSPF to multicast packets to OSPF routers. These addresses are 224.0.0.5

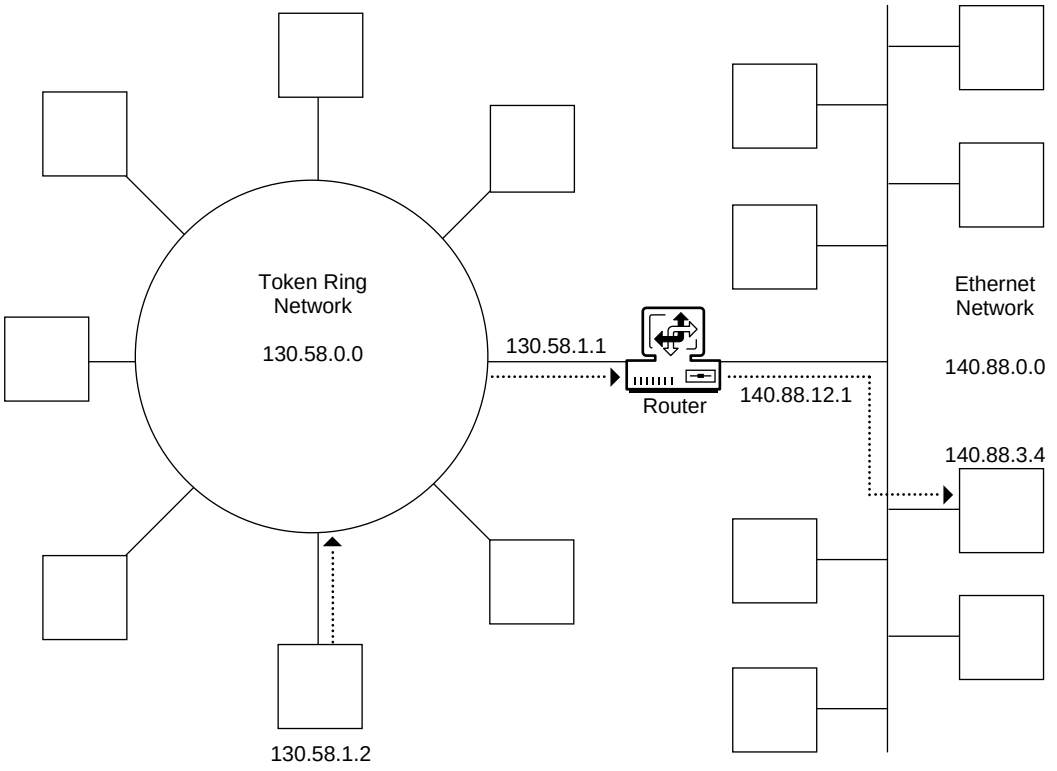
and 224.0.0.6. Two are used by Router Discovery messages to multicast router advertisements and solicitation messages. These addresses are 224.0.0.1 and 224.0.0.2. RIP II uses multicast address 224.0.0.9.

Routing

The term *routing* refers to the transmission of a datagram from one node to another on the same or a different network. The route refers to the path that is chosen to transmit an IP datagram from its origin to its destination, based on the IP addresses contained in the datagram.

When a datagram is sent to a node on another network, the network portions of the source and the destination IP addresses are different. When the packet is received by a router that connects the source to the destination network, the router forwards the packet on the correct interface to reach the destination, as shown in Figure 14 on page 36. Two networks are connected if at least one router is attached to both networks.

Figure 14 How the router connects the source to the destination network



Each host has a default router or a list of routers in other networks. When IP sends a datagram the following happens:

1. IP searches the routing table of the sending node for a default route or a path to the destination IP address.
2. IP extracts the address of the default router or next-hop router from the route entry.
3. IP requires ARP to map the next-hop address to its hardware address.
4. IP transmits the packet to the next hop.
5. IP repeats Steps 1 through 4 until the final destination is reached.

Error and Control Messages

Another protocol in the TCP/IP suite is the Internet Control Message Protocol (ICMP). ICMP packets contain information about errors and control on the network: inoperative nodes and gateways, packet congestion at a gateway, and so on. The IP software, rather than the application, interprets an ICMP message. The IP software then takes the appropriate action with respect to the ICMP message, independently of the application. Because an ICMP message might need to travel across several networks to reach its destination, it is encapsulated in the data portion of an IP datagram.

ICMP is also used to test connectivity between two nodes. The originating node uses PING to send an ICMP echo request and waits for an ICMP echo response from the destination.

Router Discovery Protocol

The Router Discovery Protocol, an extension to Internet Control Message Protocol (ICMP), allows hosts to discover routers on their networks and determine which router to use as the default router. When a host needs to send a packet to another network, it first sends the packet to a router that forwards the packet toward the destination. To accomplish this, the host needs to know where the routers are on its network and which one to send packets to.

When you configure the router discovery mechanism, the router advertises itself with periodic ICMP router advertisement messages. Then the host listens to this message and decides whether to use a router as the default router.

You can configure the host to solicit the router advertisement on attached networks. All participating routers then reply to the inquiry. By collecting those replies, the host discovers the routers on the network and determines which router to use.

A host might not select the best router (the router with the optimal path) to forward packets to a specific destination. When a router receives a packet from a host that is better forwarded to another router on the network, the router uses an ICMP Redirect message to notify the host of the optimal path.

NetWare routing software provides both host and router implementations of the Router Discovery Protocol. The mode of operation of the Router Discovery Protocol is determined by whether the IP Packet Forwarding parameter is enabled. If IP Packet Forwarding is enabled, the Router Discovery Protocol will send Router Advertisement messages. If IP Packet

Forwarding is disabled, the Router Discovery Protocol will send Router Solicitation Messages. these messages are explained in the next section.

Router Discovery Messages

The two message types that are used by the Router Discovery Protocol to communicate between hosts and routers are discussed in the following sections.

ICMP Router Advertisement Message

The ICMP Router Advertisement Message is ICMP message type 9. This message is used by routers to advertise their presence on the network and is broadcast or multicast to all hosts on the network.

This message type carries the IP address of the router and its preference level. Hosts use the preference level to determine which router to use for forwarding. The router with the highest preference becomes the default router. A value of 0x80000000 indicates the router is not to be used. Routers with this value are used *only* when other routers send ICMP Redirect messages to the host.

ICMP Router Solicitation Message

The ICMP Router Solicitation Message is ICMP message type 10. Hosts use this message to solicit router advertisements from all participating routers on the network.

Router Discovery Multicast Address

Router Discovery uses two IP multicast addresses. The IP address 224.0.0.1 is reserved to multicast the Router Advertisement Message to the hosts. The IP address 224.0.0.2 is reserved to multicast the Router Solicitation Message to the routers. If the network does *not* support multicast, then broadcast address 255.255.255.255 is used for both the Router Advertisement and Router Solicitation messages.

2 Novell TCP/IP

The Novell® TCP/IP stack consists of the following five NetWare® Loadable Module™ (NLM™) programs:

- BSDSOCK.NLM provides the BSD standards sockets interface.
- TCP.NLM provides the transport layer TCP and UDP interfaces.
- TCPIP.NLM provides IP, ICMP, IGMP, Routing and other networking layer protocols.
- NETLIB.NLM is a library of the entire stack.
- INETCFG.NLM allows you to configure the stack with the help of TCPCFG.NLM. The configuration is stored in SYS:\ETC\TCPIP.CFG and SYS:\ETC\NETINFO.CFG. Please note, an abend may cause the corruption of both of the above two .CFG files. So always take a back up of the files.

The Novell TCP/IP software is now multiprocessor (MP) enabled and multithreaded. The transport layer (TCP & UDP) is completely MP enabled so that the stack can process any TCP/UDP connections on any processor. These features are aimed at taking advantage of the multiple processors available and at making the stack scale more than what it does on a uni-processor machine.

The stack provides you with the TCP/IP protocols as per the Request For Comments. NetWare 6 gives you the following new features:

- Selective Acknowledgement (page 40)
- Large Windows (page 40)
- TCP Defend Fin Attack (page 41)
- Classless IP Addresses (page 41)

- Multihoming (page 41)
- Multiple Default Gateway (page 43)
- Dead Gateway Detection (page 43)
- Path MTU Black Hole Detection and Recovery (page 44)
- Provision of Non-ARPable Secondary IP Address (page 45)

Selective Acknowledgement

The Selective Acknowledgment (SACK) is a mechanism that includes a retransmission algorithm which helps overcome weak links on the TCP/IP stack.

The selective acknowledgment extension uses two TCP options. The first is an enabling option, SACK-permitted, which can be sent in a SYN segment to indicate that the SACK option can be used once the connection is established. The SACK-permitted option is a two-byte option.

The second option is the SACK option itself, which can be sent over an established connection once both the sender and the receiver have successfully negotiated the SACK-permit option. Whenever there is loss of data, the data receiver can send the SACK option to acknowledge the out-of-order segments.

For more information on this, see “Selective Acknowledgement” on page 57.

Large Windows

The Large Windows option allows windows larger than 2**16. It defines an implicit scale factor, which is used to multiply the window size value found in a TCP header to obtain the true window size which can go up to a maximum limit of 1 GB.

This Large Window option is negotiated while establishing the connection.

For more information on this, see “Large Windows” on page 55.

TCP Defend Fin Attack

The TCP Defend Fin Attack solution provides a simple, single tuning option, the Minimum Threshold parameter. In the TCP stack, the wait states (FIN_WAIT1, FIN_WAIT2, CLOSED_WAIT, LAST_ACK and CLOSING) are arranged in ascending order of importance by considering which of the states are less risky to terminate. The order is static.

The stack assumes that there is no risk in terminating all connections in a less important state. According to the arrangement of states, if a less important connection is over using resources then it is selected. Alternately, if an important state is over using and the less important states do not dominate, it would be selected for reset only. At any given point in time a Minimum Threshold number of connections will be permitted.

For more information on this, see "TCP Defend Land Attacks" on page 58.

Classless IP Addresses

The Novell TCP/IP stack implements Classless Inter-Domain Routing (CIDR). It is now possible to bind to supernetted addresses with non natural subnet masks. CIDR also allows binding to one or more interfaces. The NetWare system bound to a system in a supernetted IP address environment acts as an end node. In such a scenario, forwarding is disabled.

Multihoming

Multihoming enables an interface to assume multiple IP addresses on the same network. Multihoming can be used for all IP networks bound to a router. This is irrespective of whether the networks are bound to the same interface or to different interfaces. The most common use of multiple addresses on the same network is to enable a Web server to operate as though it were several Web servers. One application is to use each secondary IP address to point to a different Web page on the same Web server, depending on the Domain Name System (DNS) name that is used to reach the server.

Multihoming is also commonly used with network address translation (NAT), the proxy server, and the virtual private network (VPN). In all cases, the secondary IP address can be configured on the same interface that has the primary IP address. Or the secondary address can be configured on a different interface. When there are multiple interfaces, the secondary address is associated with the interface that is bound to the network that uses the same

address. If the secondary address is not valid on any of the networks bound to existing interfaces, the address is rejected and an error message is produced.

Load Balancing and Fault Tolerance

With this release, the multihoming feature is extended to help configure the stack for load balancing and fault tolerance at NIC/Link level. The TCP/IP stack also provides mechanisms for the administrator to group those NICs which are similar in characteristics to facilitate load balancing and fault tolerance across them.

For NetWare 6, TCP/IP has two levels of enabling load balancing and fault tolerance, one at the system level and another at the local group level. To benefit from the feature make sure that you have fully enabled it, at both the levels.

Five types of multihoming configurations are provided:

Single NIC Multiple IP Addresses (Different IP Addresses)—In this type of multihoming the client to server traffic can be distributed across the routers. The required level of load balancing can be achieved through static routers and dynamic DNS. Any dynamic algorithm would take care of the fault tolerance aspect. In the case of a static route configuration the Multiple Default Gateway with Dead Gateway Detection support would also help achieve the same results.

Multiple NIC Multiple IP Address (Different IP Network)—This is a normal configuration of a router. All the configuration and the advantages gained by the previous type of multihoming are applicable here also. In addition, load balancing server-to-client traffic across the NICs and routers is also possible. This can be achieved with the help of multipath routing.

Multiple NIC Multiple IP Address (Same IP Network)—This type of multihoming allows for both server-to-client and client-to-server load balancing and fault tolerance at the Link level. This type of multihoming is especially helpful in those cases where Route level load balancing and fault tolerance are not required and become an overhead. Once this type of multihoming is supported on the server side, the outgoing traffic load is NIC based on various parameters such as the destination IP address and interference lead. During NIC failure, the lead is automatically distributed among the remaining NICs to achieve fault tolerance.

Multiple NIC Single IP Address—Here the clients use the same IP address to communicate with the server. This is achieved through using the round robin method of distributing the NIC addresses for the ARP requests sent by the clients. This solution avoids the extra configuration and transparently achieves the load balancing for incoming packets. During a NIC failure, the server sends a message to the clients to forcefully use them as the other interface's MAC address.

Secondary IP Address (Multiple Logical Hosts)—With this type of multihoming it is possible to create multiple logical hosts belonging to the same network. In a multihoming setup where multiple NICs are grouped to support a single network, the secondary IP address with this type of configuration supports an option to select one of the NICs in the group. By using the non-ARPable option, these addresses can be used as virtual IP addresses for load balancing solutions. Through this option, the same IP address can be configured on all servers and the load balancer can distribute the client load across these servers.

For more information on this, see “Configuring Load Balancing” on page 75 and “Configuring Fault Tolerance” on page 75.

Multiple Default Gateway

This feature stretches the existing Default Gateway (Default Router) feature, by allowing you to configure multiple default gateways on your network. When a default gateway goes offline, the Dead Gateway Detection feature detects this and uses the Multiple Default Gateway list to switch to the next preferred default gateway, making your network fault tolerant.

For more information on this, see “Configuring a Default Gateway (LAN Static Route)” on page 71.

Dead Gateway Detection

This feature is used with the Multiple Default Gateway feature. When the current default gateway goes offline, this feature detects the failure and automatically enables the next preferred default gateway from the Multiple Default Gateway list to act as the current default gateway. When a dead default gateway with a higher preference is again online, this feature detects this and switches back to the default gateway with the higher preference.

For more information on this, see “Enabling Dead Gateway Detection” on page 74 and “Configuring Dead Gateway Detection” on page 74.

Probe Interval

This configured parameter lets you fine-tune the performance of the Dead Gateway Detection feature by modifying the time interval (in seconds) at which probes are sent to the default gateway to determine whether it is functional or not.

For more information on this, see “Configuring Probe Interval” on page 74.

Probe Timeout

This configured parameter sets the time interval (in seconds) after which the next probe is sent to the default gateway, when there is no reply received by the gateway for the previously sent probe.

For more information on this, see “Configuring Probe Timeout” on page 75.

Path MTU Black Hole Detection and Recovery

This feature provides the facility to detect a connection failure due to black hole routers and helps to recover such connections.

Whenever a router gets a datagram with Don't Fragment (DF) bit set in its header and the packet size is greater than the next MTU the router cannot forward the packet. In such a case, the router sends an ICMP Destination Unreachable DF bit set message to the host.

Often routers do not send such a message. Instead they ignore the datagram. Typically, an IP datagram cannot be forwarded because its maximum segment size is too large for the receiving server and the Don't Fragment bit is set in the header of the datagram. Routers that ignore these datagrams and send no message are called PMTU black hole routers. Some routers might silently drop large frames, even when the DF bit is not set. Firewalls are often misconfigured to suppress all ICMP messages.

To respond effectively to black hole routers, the Novell TCP/IP stack now provides a Path MTUBH Detect feature. Path MTUBH Detect recognizes repeated unacknowledged transmissions and responds by turning off the Don't Fragment bit. After a datagram is transmitted successfully, the MTUBH Detect feature reduces the maximum segment size and turns the Don't Fragment bit on again.

The feature specifies the maximum transmission unit size of an interface. Each media type has a maximum frame size that can't be exceeded. The Link layer is responsible for discovering this MTU and reporting it to the protocols above it.

For more information on this, see “Path MTU Black Hole Detection and Recovery” on page 57.

Provision of Non-ARPable Secondary IP Address

This feature lets you add a secondary IP address which will not reply to any of the ARP requests coming from the network.

For more information on this, see “ARP Cache Stale Timeout” on page 53 and “ARP Cache Update Timeout” on page 54.

3 Setting Up

This chapter describes how to set up the basic components of Novell® TCP/IP. For this release, only LAN configurations are supported.

The following topics are discussed:

Configuring Boards

Configuring, or reconfiguring, a board involves choosing a driver for the board, assigning a name to the board, and configuring the board parameters.

When you select and configure a LAN board, you are actually configuring one or more physical interfaces that correspond to individual connections over which packets are routed. Configuring a board causes the driver associated with the board to load each time you initialize the router.

Most drivers that are compatible with NetWare software have a *driver description file* that defines the hardware parameters necessary for the driver to operate with the board you select. This file—sometimes called the *.LDI file* — also specifies the valid range of values for each parameter. If a driver has an .LDI file, the parameters associated with that driver are presented in the Board Configuration menu; you simply choose a value for each parameter. If a driver has no.LDI file, you must enter the required values in the Board Parameters field.

Loading INETCFG

To load INETCFG at from the server prompt enter

```
inetcfg
```

The Internetworking Console interface is displayed. For the next set of operation you need to select options on the Internetworking Console screens.

Configuring a LAN Board

To configure a board, complete the following steps:

1. Load INETCFG and then click Boards.
2. Do one of the following:

   2a If you are configuring a new board:

   - Press Ins to display the list of available drivers.
   - Scroll through the list of available drivers and select the driver that corresponds to the type of new LAN board you are installing in your system. If the driver you need is not in the list, refer to "Adding a New Board Driver or NLM File to Your System" on page 50.

   2b If you are changing an existing board configuration:

   Select that board.

   - Press Enter to see the configured parameters of the board.
   - Change the required parameters. (The name can not be changed.)

3. The Configured Boards screen is displayed.

   NOTE: If you are doing a new configuration, no existing boards are shown. Otherwise, boards that have already been configured are shown.

   The Configured Boards screen displays a list of configured boards with some or all of the following information:

   - Board Name—Name you assign to the board.
   - Driver—Name of the driver associated with the board.
   - Int—Interrupt request level (IRQ) used by the board.
   - IOAddr—Base input/output port address for the board.
   - MemAddr—Base memory address used by the board.

- Slot—Number of the slot where the board is installed.
- Status—Status of the board, which is Enabled by default.
- Comment—Any comments that you enter about the board or its configuration.

NOTE: Not every board-driver configuration requires all this information; in fact, some configurations require other, link-specific parameters that are not shown in the Configured Boards screen. These parameters are displayed in the Board Configuration menu, as described in the following steps.

If the board driver has an .LDI file, the parameters you need to configure for the board are displayed as separate fields in the menu.

If the board driver has no .LDI file, only the Board Name, Board Parameters, and Comment fields are provided as a means for entering the parameters manually.

4. Specify the board parameters by doing one of the following:
   - If the driver selected has a description file, the parameters are listed as separate fields. You must highlight each field one at a time and select the appropriate value for the parameter from the displayed list.

     HINT: Use the context-sensitive help text if you need an explanation of any parameter. Highlight the parameter and press F1 to display the help text. Press Esc to exit the help screen. When in doubt, accept the default values.
   - If the driver selected does not have a description file, the Board Configuration Without A Driver Description File menu is displayed. You must type the parameters in the Board Parameters field; use the following as an example:

     ```
     PORT=300 INT=3
     ```

     These parameters are appended to the **LOAD** *driver* line.
5. Press Esc to return to the Configured Boards screen; save your changes when prompted.

   The Configured Boards screen now shows the board you just configured. Note that the board status is Enabled; you can use the Tab key to toggle between Enabled and Disabled. To ensure that the board is loaded, continue with the next step.
6. Press Esc to return to the Internetworking Configuration menu; save your changes when prompted.
7. If you want these changes to take effect immediately, perform the following actions.

LAN boards with a single network interface need no further configuration; however, an enabled driver is not loaded unless a protocol is bound to it.

After a data-link protocol has been associated with the board, select Reinitialize System or restart the router to make the changes take effect. If there are any conflicts with the hardware parameters of other boards, one or more messages describe them. You must determine whether the conflicts are acceptable or whether they interfere with the operation of the router and, if necessary, resolve them.

Adding a New Board Driver or NLM File to Your System

1. Load INETCFG and then click Boards.
2. Press Ins to display the list of available drivers.
3. Press Ins again.

   All the available Novell certified drivers will appear on the screen.
4. Select the driver and then press Enter.

   NOTE: You can also use this screen for copying drivers and NLM files from a floppy diskette to the SYS:/SYSTEM directory.

   To add a driver to the list of available drivers, insert the diskette containing the driver in one of the disk drives, specify the complete path and filename of the driver (for example, A:\NEWDRV\DRIVER) and then press Enter. The driver and its description file, if any, are copied into the SYS:SYSTEM directory. (Note that the file extension is not required.)
5. Configure the new board as described in "Configuring a LAN Board" on page 48.

Enabling or Disabling a LAN Board

1. Load INETCFG and then click Boards.
2. Select the board you want to enable or disable and press Tab.

   The screen displays the board's new status (Enabled or Disabled).

   IMPORTANT: If you disable a board that uses an AIO or CAPI driver and you reinitialize the system, then all other boards that use the same driver will also be disabled. If this happens, you must restart the server to reload all instances of the driver that were loaded for another product in AUTOEXEC.NCF (without INETCFG). However, the board loaded in AUTOEXEC.NCF will remain inoperable until you restart the server or until you enter the LOAD command at the console

and reinitialize the system. To avoid this problem, use INETCFG to load both drivers.

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

Deleting a LAN Board

1. Load INETCFG and then click Boards
2. Select the board you want to delete and press Del.

   A message is displayed indicating that deleting the board also deletes all existing binds to the board's interfaces.
3. When prompted, select Yes to delete the board.

   The board is removed from the list of configured boards.
4. Press Esc to return to the Internetworking Configuration menu.
5. Reinitialize system for the changes to take place.

Enabling TCP/IP

1. Load INETCFG and then click Protocols.
2. Select TCP/IP from the list of protocols.
3. In the TCP/IP screen, enable or disable the following:
   - TCP/IP
   - IP Packet Forwarding
   - OSPF
   - LAN Static Routing
   - Dead Gateway Detection
   - SNMP Manager
   - DNS Resolver
   - Load Balancing
   - Fault Tolerance
   - Filter Support
   - NAT Implicit Filtering
   - Expert Configuration

You can further set the detailed configuration of each of the above fields. For more information of how to configure TCP/IP, see chapter Chapter 3, “Setting Up,” on page 47.

Binding Protocols

1. Load INETCFG and then click Binding.
2. Select a protocol.

   Here you need to enter data in terms of Protocol, Interface/Group, Status and Identifier. There might be instances when you need to enter data in the screens still to come and that is reflected here.
3. Enter the relevant parameter information in the Binding TCP/IP to a LAN interface screen and then click Configure TCP/IP Bind options.
4. Either set both the Group Interface for LB/FT and Set as Primary Interface to Yes or set the first one to No.

   Setting the first to No enables RIP Bind Options, OSPF Bind Options and Expert TCP/IP Bind Options. Choose the relevant screen and enter the data into it.

4 SET Parameters

This chapter discusses how to use SET parameters to change some of the default parameters or enable/disable the various features provided in Novell® TCP/IP stack.

Configuration Using SET Options

The following SET options allow you to configure certain parameters from the command line on the host. The SET options are entered at the server console as commands, and the configuration changes made this way are applied to the whole system rather than to an individual interface.

ARP Cache Stale Timeout

Syntax	set arp entry expiry time = *n*
Description:	Specifies a timeout period (in seconds) for a cache table entry to be removed from the ARP cache table if the entry has not been used for some time.
Range:	240 to 14400 (seconds)
Default:	300

ARP Cache Update Timeout

Syntax:	set arp entry update time = *n*
Description:	Specifies a timeout period for a cache table entry to be removed from the ARP cache table if the entry has not been updated.
Range:	240 to 14400 (seconds)
Default:	300

BSD Socket Default Buffer Size

Syntax:	set bsd socket default buffer size in bytes = *n*
Description:	Sets the BSD Socket default send and receive buffer sizes.
Range:	4096 to 65536 (bytes)
Default:	32768

Discard Oversized Ping Packets

Syntax:	set discard oversized ping packets = *string*
Description:	Enable or disable discarding ping packets of size larger than the current ping packet size that is set to be received.
Range:	On | Off
Default:	On (enabled)

Discard Oversized UDP Packets

Syntax:	set discard oversized udp packets = ***string***
Description:	Enable or disable discarding UDP packets of size larger than the current ping packet size that is set to be received.
Range:	On | Off
Default:	On (enabled)

IP Address Duplicates

Syntax:	set allow ip address duplicates = *string*
Description:	Binds the IP address even if it conflicts with another node in the network. (TCP/IP normally will not allow you to bind IP addresses that conflict with other nodes in the network.)
Range:	On | Off
Default:	Off (disabled)

Large Windows

Syntax:	set tcp large window option = *string*
Description:	Enable or disable the Large Window option.
Range:	On | Off
Default:	On (enabled)

Largest Ping Packet Size

Syntax:	set largest ping packet size = ***n***
Description:	Set the size of the largest ping packet that can be received.
Range:	0 to 65535 (bytes)
Default:	10240

Largest UDP Packet Size

Syntax:	set large udp packet size = ***n***
Description:	Set the size of the largest UDP packet that can be received.
Range:	0 to 65535 (bytes)
Default:	16384

Maximum Packet Receive Buffers

Syntax:	set maximum packet receive buffers = *string*
Description:	Allocate the maximum packet receive buffers to the server. This option can be set in the AUTOEXEC.NCF file. You can also use this option to fine-tune the server when it is stressed.
Range:	50 to 3303820 (packets)
Default:	10000

HINT: If you want to see the current value, do not assign any variable to set the parameter.

Maximum Pending TCP Connection Requests

Syntax:	set maximum pending tcp requests = ***n***
Description:	Set the maximum number of pending TCP connections.
Range:	128 to 4096
Default:	128

Minimum Packet Receive Buffers

Syntax:	set minimum packet receive buffers = *string*
Description:	Allocate the minimum packet receive buffers to the server. This option can be set in the AUTOEXEC.NCF file. You can also use this option to fine-tune the server when it is stressed.
Range:	10 to 32768 (packets)
Default:	2000

Path MTU Black Hole Detection and Recovery

Syntax:	set tcp path mtu black hole detection and recovery = *string*
Description:	Enable or disable the Path MTU Black Hole Detection and Recovery option.
Range:	On | Off
Default:	Off (disabled)

Selective Acknowledgement

Syntax:	set tcp sack option = *string*
Description:	Enable or disable the SACK option.
Range:	On | Off
Default:	On (enabled)

TCP Diagnostic Services

Syntax:	set tcp diagnostic services = *string*
Description:	Enable or disable the *echo*, *discard* and *chargen* TCP diagnostic services on the NetWare® system.
Range:	On | Off
Default:	Off (disabled)

TCP Defend Fin Attacks

Syntax:	set maximum wait states = ***n***
Description:	Enable or disable defense against Fin attacks.
Range:	1 to 100000
Default:	0 (disabled)

TCP Defend Land Attacks

Syntax:	set tcp defend land attacks = *string*
Description:	Enable or disable defense against land attacks.
Range:	On | Off
Default:	On (enabled)

TCP Defend SYN Attacks

Syntax:	set tcp defend syn attacks = *string*
Description:	Enable or disable defense against SYN attacks.
Range:	On | Off
Default:	Off (disabled)

TCP IP Maximum Small ECBs

Syntax:	set tcp ip maximum small ecbs = *string*
Description:	Change the maximum number of small ECBs on the server. You can also use this option to fine-tune the server when it is stressed.
Range:	512 to 65534 (packets)
Default:	1024

TOS Value

Syntax:	set tos for ip packets = *n*
Description:	Specifies a TOS value for all the outgoing IP datagrams through this interface. Assign a value to set the TOS and the precedence bits of the IP header for outgoing packets. To set only TOS bits, use a value between 0 and 15. To set the TOS and the precedence bits, use a value between 0 and 127.
Range:	0 to 127
Default:	0

5 Protocols

The Novell® TCP/IP software provides a set of configurable parameters with which you can modify operational characteristics of the Internet Protocol (IP). You can select its routing protocol and configure it to run over a LAN. In NetWare® 6, configuration over WAN connections is not supported.

To configure IP, you enable the protocol, set its parameters, and bind it to a network interface. You configure all IP parameters from the Internetworking Configuration utility (INETCFG).

To understand what decisions must be taken before you configure TCP/IP beyond its most basic configuration, refer to Appendix A, "Planning," on page 103.

NOTE: The configuration you specify with INETCFG does not take effect automatically. To activate the configuration, save your changes and press Esc until you see the Internetworking Configuration menu. You can then select Reinitialize System > Yes to activate your changes.

The following are discussed in this chapter:

Configuring RIP

RIP is probably the most common IP routing protocol in use. It is widely available and presents few obstacles to interoperability with other IP internetworks, most notably the Internet.

RIP performs sufficiently well in small IP internetworks that have simple architectures and few routers. However, RIP reveals its limitations in the large, complex internetworks that have become common in government and private-sector organizations throughout the world. Its most apparent limitations are the following:

- All subnets must be contiguous
- RIP routes are limited to 15 hops

To overcome or ease some of these limitations, the internetworking community developed various enhancements to RIP. RIP II, for example, is an enhanced version of RIP that supports variable-length subnet masks. It carries a field that contains the subnet mask of the destination network. RIP II also supports the use of subnet zero, whose addresses were reserved under the original IP specification. When configuring RIP on your router, you can run RIP I, RIP II, or both on a single interface.

NOTE: Not all third-party routers support RIP II.

You can also enable *poison reverse* on an interface. This is a mechanism that causes RIP to advertise a route back through the same path from which it learned the route, but with a hop count of 16—that is, unreachable. Although poison reverse prevents routing loops, the unreachable routes carried in each RIP packet increase the bandwidth consumed by RIP traffic. This increase becomes significant in large internetworks.

RIP enables you to assign a *cost* value between 1 and 15 to each network interface you configure. This enables you to establish a preferred route according to the type of network media connected to the interface. For

example, you might want to increase the cost of an interface that uses a slow link so that, given the choice, RIP uses the interface to a faster, less costly link. The default cost for each interface is 1. Do not increase this value on an interface unless you want to discourage its use as an eligible routing path.

When choosing an IP routing protocol, consider the following guidelines:

- If the IP internetwork is small and uses no routing protocol besides RIP, continue using RIP.

  To configure RIP on the router, see "Configuring RIP" on page 60.

  However, if the network will continue to grow and perhaps become part of a larger IP internetwork, you should consider migrating the network from RIP to OSPF.

- If the internetwork uses variable-length subnets or has third-party routers that support RIP II, use RIP II or OSPF.

  To configure RIP II, see "Configuring RIP" on page 60. To configure OSPF, see "Configuring OSPF" on page 63.

- If the internetwork has some third-party routers that support RIP II and others that do not, use RIP I *and* RIP II.

  For instructions on enabling RIP I and RIP II simultaneously on a network interface, see "Configuring RIP" on page 60.

- If you are currently building a large IP internetwork, use OSPF.

  You can also run RIP and OSPF concurrently; for more information, see "Configuring OSPF" on page 63.

  To enable RIP routing on the router and to configure RIP on a network interface, do the following:

1. Load INETCFG and then select Protocols > TCP/IP
2. Make sure RIP routing is enabled globally by setting the RIP field to Enabled.

   This is the default setting.

   If you want to disable RIP routing on a single interface, set the Status parameter in the RIP bind options to Disabled. This action is described in Step 3.
3. Press Esc twice to return to the Internetworking Configuration menu and then select Bindings > *an existing binding* > RIP Bind Options.

   Configure the following parameters:

- Status—Status of RIP routing on this interface. RIP routing is enabled by default; to disable RIP routing only on this interface, select this parameter, then select Disabled.
- RIP Version—Version of RIP to use on this interface. Select one of the following options:

  RIP I—Standard version of RIP used by most IP routers and end nodes. This is the default option.

  RIP I & RIP II—Both versions of RIP. Select this option if your internetwork has nodes that support both RIP I and RIP II.

  RIP II—Enhanced version of RIP that supports variable-length subnet masks.
- RIP Mode—Mode of the RIP version you selected in RIP Version.

  Normal—Causes the router to send and accept RIP packets, RIP I, RIP II, or both.

  Receive Only—Causes the router to only receive RIP packets.

  Send Only—Causes the router to broadcast, in RIP packets, only the entries in its own routing table.

  Some end nodes learn routes only by listening to RIP, even if portions of the internetwork run OSPF. Select Send Only if you want the router to broadcast the OSPF routes in its RIP I packets so that every end node can learn *all* available routes.

The RIP Bind Options menu also includes the following parameters:

- Cost of Interface
- Originate Default Route
- Poison Reverse
- Split Horizon
- Update Time
- Expire Time
- Garbage Time
- RIP II Options

IMPORTANT: Because the default settings for these parameters are suitable for most IP networks, you should change them only for a specific purpose. Incorrectly

configuring these parameters can increase routing traffic or cause loss of connectivity on your network.

4. Press Esc until you are prompted to save your changes, and then select Yes.
5. Press Esc to return to the Internetworking Configuration menu.
6. If you want these changes to take effect immediately, select Reinitialize System > Yes to activate your changes.

Configuring OSPF

OSPF was developed to satisfy the need for a scalable, open-standards routing protocol for large IP internetworks. It is a *link state* protocol that provides highly efficient routing and fast convergence.

OSPF makes large internetworks more manageable by enabling you to partition them into administrative domains called *areas*. Areas impose a hierarchy to the internetwork. All OSPF areas are connected to a central *backbone* area by an *Area Border Router* (ABR). The ABR shares OSPF routing information between the area and the backbone.

When configuring an OSPF area, you assign to it a 4-byte decimal number called the *Area ID*. You also indicate which of the router's network interfaces belong to the area and whether the area is a *stub area*.

Novell TCP/IP supports the use of *virtual links* between OSPF routers. A virtual link patches together a partitioned backbone. It creates a direct point-to-point link between the ABRs that connect the partitioned backbone areas through the *transit area*.

Most IP internetworks in use today are not pure OSPF networks; that is, portions of these internetworks still employ other routing protocols, such as RIP. OSPF uses an *Autonomous System Boundary Router* (ASBR) to import and propagate routing information from these protocols. ASBRs are always located on the border of an OSPF domain. When configuring OSPF, you can enable your router to operate as an ASBR. For an ASBR to import RIP routes learned through an interface, RIP must be enabled on that interface.

Each OSPF router has its own *Router ID*, a 4-byte number that uniquely identifies the router and enables it to participate in informational exchanges with neighboring routers. The default Router ID is the IP address of the first interface bound to IP on the router. Although INETCFG enables you to change

the Router ID, you should use the default unless you need a simpler numbering scheme for administrating several hundred routers on an internetwork.

HINT: If you are using an unnumbered point-to-point interface, we recommend that you configure a unique router ID.

Optionally, OSPF can be configured to *authenticate* its packets by providing an *authentication key* —an 8-byte, alphanumeric password—in each OSPF packet header. OSPF authentication gives you administrative control over which routers participate in link state exchanges on the internetwork. A router without proper authentication is excluded from these exchanges and, essentially, from performing any OSPF routing whatsoever. Novell TCP/IP enables you to provide authentication for an area and to provide an authentication key for each network to which the router is connected. By default, authentication is turned off.

OSPF enables you to assign a *cost* value to each network interface you configure. This enables you to establish a preferred route according to the type of network media connected to the interface. For example, you might want to increase the cost of an interface that uses a slow link so that, given the choice, OSPF uses the interface to a faster, less costly link.

Like RIP, OSPF can run over most WAN connections, depending on which call type you use. On-demand calls, for example, typically use static routes instead of an active routing protocol.

IMPORTANT: An active routing protocol, such as OSPF, should not be used on an on-demand link because it will periodically bring up the link and will cause the link to continue to stay up.

Permanent calls on an IP network typically use a routing protocol, such as OSPF or RIP, to communicate routing information. However, they can also use static routes to conserve bandwidth. OSPF can also run over a nonbroadcast multiaccess network, such as X.25 or frame relay, but you must provide the IP address of the peer OSPF router at the other end of each connection.

HINT: Novell TCP/IP enables you to run OSPF and RIP on the same router, but under normal circumstances, you should run them separately on different interfaces. Although an ASBR must run both protocols so that it can import RIP routes and propagate them to other OSPF routers, you should not run both on too many other routers in your OSPF domain. Doing so consumes additional network bandwidth and router memory, and might even create routing loops.

The extent to which you must configure OSPF depends on the characteristics of your network, such as its size and topology, and whether it uses other IP

routing protocols besides OSPF. To help you configure only what is necessary, this section provides the following procedures:

- Basic OSPF configuration
- Advanced OSPF configuration

Basic OSPF Configuration

To enable OSPF routing on the router and to configure OSPF on a network interface, do the following:

1. Load INETCFG and then select Protocols > TCP/IP
2. Select the OSPF field and then Enabled.

   This action enables OSPF routing globally on the router. If you want to disable OSPF routing on a single interface, set the Status parameter to Disabled as described in Step 3.
3. Press Esc repeatedly to return to the Internetworking Configuration menu and then select Bindings > *an existing binding* > OSPF Bind Options.

   The Status field indicates whether OSPF routing is active on this interface. OSPF routing is enabled by default; to disable OSPF routing only on this interface, select Status, then select Disabled.

   The OSPF Bind Options menu also includes the following parameters:

   - Cost of Interface
   - Area ID
   - Priority
   - Authentication Password
   - Hello Interval
   - Router Dead Interval
   - Neighbor List

   IMPORTANT: Because the default settings for these parameters are suitable for most IP networks, you should change them only for a specific purpose. Misconfiguring these parameters can increase routing traffic or cause loss of connectivity on your network.
4. Press Esc until you return to the Internetworking Configuration menu. Select Yes if you are prompted to save your changes.

5 If you want these changes to take effect immediately, select Reinitialize System > Yes.

Advanced OSPF Configuration

To configure advanced OSPF features, do the following:

1 Load INETCFG and then select Protocols > TCP/IP.

2 Select OSPF Configuration.

The OSPF Configuration menu is displayed and includes the following parameters:

- Router ID
- Virtual Link Configuration
- IP Load Sharing

IMPORTANT: Most network configurations do not require you to change these parameters.

3 To configure an ASBR, select Autonomous System Boundary Router > Enabled.

Enabling this parameter enables the router to operate as an ASBR. In this capacity, the router advertises non-OSPF routes, such as those generated by RIP and EGP. In addition, static routes and direct routes to the OSPF domain are advertised. This is necessary to preserve connectivity throughout an internetwork that uses routing protocols other than OSPF. This parameter should be configured only on routers that connect an OSPF area to an area that uses a different routing protocol.

Do not enable this parameter on an internetwork that uses only OSPF. Doing so causes unwanted traffic on the route.

4 To configure an OSPF area, select Area Configuration and continue with Step 5. Otherwise, go to Step 11.

The OSPF Areas menu is displayed.

This menu lists the IDs of all areas to which the router belongs. If you have not configured an OSPF area on this router, the only area listed is 0.0.0.0, the *backbone area*.

5 Select an existing area or press Ins to create a new area.

6 Configure the following area parameters:

- Area ID—Four-byte decimal number that identifies the area. For example, a valid Area ID is 85.8.0.11. However, the Area ID does not have to be an IP address. You can enter any number, but it must be in the format of an IP address. If you enter a hexadecimal number, INETCFG converts it to decimal.

  For the router to belong to an area, the Area ID that identifies that area must be assigned to at least one of the router's interfaces. You assign an Area ID to an interface in Step 8.

- Authentication—Switch that enables or disables authentication for the area.

  If you enable authentication on this router, you must enable authentication on all other routers in the area. Also, all interfaces belonging to that area must have an *authentication key*. You provide the authentication key in Step 8.

- Route Aggregation—Network number of a group of networks that is aggregated into one network number. Press Ins to assign the Network and Mask values of this network number. Because supernetting is not supported, the aggregated network must be the same length as the natural mask of the network class.

- Area Type—Type of OSPF area, which can be Normal or Stub. All routers in the same area must agree on the area type.

  NOTE: The backbone area (0.0.0.0) cannot be a stub area.

- Stub Cost—Cost of the default route advertised to the stub area. This parameter is used only if the Area Type is set to Stub.

7 Press Esc until you are prompted to save your changes, and then select Yes.

8 Press Esc until you return to the Internetworking Configuration menu, then select Bindings > *an existing binding* > OSPF Bind Options.

9 If you are configuring an OSPF area, configure the following area parameters:

- Area ID—ID of the area to which this interface belongs. Press Enter to determine the list of available areas. Use the Up-arrow and Down-arrow keys to select an area, and then press Enter to select it.

- Authentication Password—Eight-byte password that authenticates the router's OSPF packets to the area to which this interface belongs. Valid characters are 0 to 9, A to Z, a to z, underscore, and dash.

This parameter is required only if you enabled the Authentication parameter for the area you select, as described in Step 6 on page 66.

IMPORTANT: Not all interfaces within the same area are required to have the same authentication key; however, all interfaces *connected to the same network* must have the same authentication key.

10 Press Esc until you are prompted to save your changes, and then select Yes.

11 Press Esc to return to the Internetworking Configuration menu.

12 If you want these changes to take effect immediately, select Reinitialize System > Yes.

Configuring Load Sharing over Equal-Cost OSPF Routes

IP maintains multiple equal-cost OSPF routes. Load sharing enables a router to divide traffic over equal-cost routes. The router can have several next hops available toward any destination. With this configuration, the router can divide the traffic among the various equal-cost routes to the destination. As a result, load sharing increases the effective bandwidth of an end-to-end path. In addition, it can improve the traffic distribution on an internetwork.

NOTE: Load sharing is performed only on equal-cost routes learned from OSPF.

You enable load sharing within OSPF. IP maintains a maximum of four equal-cost routes to each destination network. The OSPF equal-cost routes are maintained internally and are not displayed in TCPCON.

IMPORTANT: Because OSPF networks tend to be large and complex, we recommend that you do not manually adjust the cost of the interface to create equal-cost routes. It is best to let OSPF automatically determine the equal-cost routes to the destination network.

To configure load sharing on the router do the following:

1 Load INETCFG, and then select Protocols > TCP/IP.

2 Select OSPF > Enabled.

3 Select OSPF Configuration.

4 Select IP Load Sharing > Enabled.

5 Press Esc until you are prompted to save your changes, and then select Yes.

6 Press Esc to return to the Internetworking Configuration menu.

7 If you want these changes to take effect immediately, select Reinitialize System > Yes.

Configuring Static Routes for LANs

Static routes are useful if you want to do any of the following on your network:

- Eliminate routing traffic, which increases the bandwidth available for data.
- Limit user access to one portion of the network. For example, if a static route for a network is configured on a router, any packets that are received by the router are forwarded only to the destination network specified by that static route.
- Gain access to isolated areas of the network, which is useful if dealing with legacy network topologies.
- Gain access to a network more than 15 hops away.
- Use a static route as a backup route to dynamic routes.

HINT: Use this procedure to configure static routes when the next hop router is on the same LAN as the router you are configuring.

How to Configure a LAN Static Route

To configure a static route for a LAN do the following:

1 Load INETCFG, and then select Protocols > TCP/IP

2 Configure the following static route parameters:

- LAN Static Routing—Enables LAN static routing on the router.

  Select this field, and then select Enabled.

- LAN Static Routing Table—Entry point to the LAN static route configuration parameters.

  Press Ins and configure the following parameters:

  Route Type—Select Network or Host if you want the destination at the other end of the static route to be a single IP host or an IP network (that is, a group of hosts). Or, you can select Default Route. If the router must forward a packet for which it can find no destination in its routing table, it sends the packet to the address specified by the next hop for the default route. This type of blind forwarding keeps a

packet on the network until a router can forward it to its final destination.

IP Address of Network/Host—Enter the address of the destination network or host. To select from a list of symbolic network or host names and addresses, press Ins. The list of symbolic network names and addresses comes from the SYS:\ETC\NETWORKS file. The list of symbolic host names and addresses comes from the SYS:\ETC\HOSTS file.

Subnetwork Mask—If the destination is an IP network, the subnet mask of that network.

Next Hop Router on Route—Explicit destination of the next hop.

Enter the IP address of the next-hop router. To select from a list of symbolic hostnames and addresses, press Ins.

Metric for This Route—Number of hops to the destination. This metric is directly proportional to the cost of the route. Given two routes to the same destination, the router chooses the lower-cost route.

If you want to use the static route as a *backup route* to a dynamic route, select a value that is higher than the cost associated with the dynamic route. This selection ensures that the dynamic route remains the preferred route under typical conditions.

Do not set this metric value to 16 unless you want to disable the route.

Type of Route—Specify whether the static route is *active* or *passive*. This parameter specifies whether the next hop router for this route actively advertises the route to this network.

Usually, static routes are not advertised and are categorized as passive routes. When a route is marked as active, TCP/IP expects the next hop router to advertise the route regularly. If a router stops advertising an active route, TCP/IP assumes the route is no longer available and deletes it from the routing table.

If the static route is active and the router discovers a lower-cost dynamic route to the same destination, it uses the lower-cost route instead of the active static route. If the lower-cost route becomes unavailable, the router returns to using the active static route.

3 Press Esc twice, and then select Yes to save your changes.

4. Optional: Disable the routing protocol on this interface to reduce routing traffic.

   4a. Select Bindings > *an existing binding*.

   4b. Select RIP Bind Options > Status > Disabled

   4c. Press Esc and then select OSPF Bind Options > Status > Disabled

   4d. If your router has multiple interfaces and you want to disable them, repeat these steps for each interface.

5. Press Esc until you are prompted to save your changes, and then select Yes.

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

7. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Configuring a Default Gateway (LAN Static Route)

You can configure multiple default gateways and provide a fault tolerant solution on your network. When the Dead Gateway Detection feature is enabled and a default gateway goes off-line, the Multiple Default Gateway list is used to switch to the next preferred default gateway, thereby reducing the downtime of your network.

To configure the Dead Gateway Detection, see "Configuring Dead Gateway Detection" on page 74. The various ways of configuring multiple default gateways are explained below. The implications of using different methods for configuring multiple default gateways are also given in Table 3 on page 73.

Using INETCFG to Add a Default Gateway

To configure multiple default gateways using Internetworking Configuration, see "How to Configure a LAN Static Route" on page 69.

Using BIND to Add a Default Gateway

To add a new default gateway using BIND, enter the following command at the server prompt:

```
bind ip board name addr=x.x.x.x mask=x.x.x.x
gate=x.x.x.x cost=n
```

board name is the name of the interface board that you assigning as the default gateway. *addr* is the IP address of the board. *mask* is the subnet mask address associated with the IP address of the board. *gate* is the IP address of the gateway, and *cost* is the value associated with the gateway and is also the primary routing metric for this gateway.

Using TCPCON to Add a Default Gateway

1. Load TCPCON and then select IP Routing Protocol.
2. Select Proceed, press Ins, and configure the following parameters:

   Destination—Press Ins to display a list of symbolic network names from the SYS:\ETC\NETWORKS file. Select Default here.

   Next Hop—Enter the IP address of the gateway.

   Interface—Enter the interface index value through which the next hop of this gateway should be reached.

   Cost—Enter the primary routing metric for this gateway.
3. Press Esc until you are prompted to save your changes, and then select Yes to return to the IP Routing Table screen.

Configuring RIP to Add a Default Gateway

1. Load INETCFG and then select Bindings > *an existing TCP/IP binding* > RIP Bind Options.
2. Select Originate Default Route and enable this option.
3. Press Esc until you are prompted to save your changes, and then select Yes.
4. Press Esc to return to the Internetworking Configuration menu.
5. If you want these changes to take effect immediately, select Reinitialize System > Yes.

These advertisements from the router are picked up and used by an end node to add a default gateway on it. This is possible only when RIP is enabled on the end node.

Configuring Router Discovery to Add a Default Gateway

You can configure Router discovery options on a system to add a default gateway. These systems advertise themselves as a gateway and answer queries

from end nodes. End nodes use it to locate a gateway on their network. To configure Router Discovery, see "Configuring Router Discovery" on page 76.

Comparing Different Default Gateway Configuration Methods

The table below explains the implications of using different methods for configuring multiple default gateways and describes the priorities associated with each method.

Table 3 Comparison of Default Gateway Configuration Methods

Configuration Method Used to Add Multiple Default Gateways	Metric or Cost Assigned to Default Gateway		
	Better Than the Cost of Current Default Gateway	Equal to the Cost of Current Default Gateway	Worse Than the Cost of Current Default Gateway
Using INETCFG to Add a Default Gateway (page 71)	Will be added to the Default Gateway list and will be made the default gateway.	Will not be made the default gateway, but entry will be added to the Default Gateway list.	Will not be made the default gateway, but entry will be added to the Default Gateway list.
Using BIND to Add a Default Gateway (page 71)	Will be added to the Default Gateway list and will be made the default gateway.	Will not be made the default gateway, but entry will be added to the Default Gateway list.	Will not be made the default gateway, but entry will be added to the Default Gateway list.
Using TCPCON to Add a Default Gateway (page 72)	Will be added to the Default Gateway list and will be made the default gateway.	Will be added to the Default Gateway list and will be made the default gateway.	Will not be added to the Default Gateway list and will not be made the default gateway.
Configuring RIP to Add a Default Gateway (page 72)	Will be added to the Default Gateway list and will be made the default gateway.	Will not be added to the Default Gateway list and will not be made the default gateway.	Will not be added to the Default Gateway list and will not be made the default gateway.
Configuring Router Discovery to Add a Default Gateway (page 72)	Has the least priority. Will be added to the Default Gateway list but will not be made the default gateway until the default gateways added through INETCFG, TCPCON, BIND, or RIP are absent.	Has the least priority. Will be added to the Default Gateway list but will not be made the default gateway until the default gateways added through INETCFG, TCPCON, BIND, or RIP are absent.	Has the least priority. Will be added to the Default Gateway list but will not be made the default gateway until the default gateways added through INETCFG, TCPCON, BIND, or RIP are absent.

Enabling Dead Gateway Detection

1. Load INETCFG and then select Protocols > TCP/IP.
2. Select Dead Gateway Detection > Enabled.
3. Press Esc until you are prompted to save your changes, and then select Yes.
4. Press Esc to return to the Internetworking Configuration menu.
5. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Configuring Dead Gateway Detection

You can fine-tune the performance of the Dead Gateway Detection feature.

Probe Interval is the time (in seconds) at which probes would be sent to the default gateway to determine whether it is functional or not.

The valid range of values for Probe Timeout is from 10 to 1800 seconds. Default value is 30 seconds.

Probe Timeout is the time interval (in seconds) after which the next probe is sent to the default gateway, when there is no reply received by the gateway for the previously sent probe.

The valid range of Probe Timeout values is from 1 to 20 seconds. Default value is 2 seconds.

Configuring Probe Interval

1. Load INETCFG and then select Protocols > TCP/IP > Dead Gateway Detection Configuration.
2. Select Probe Interval and then enter a value in seconds.
3. Press Esc until you are prompted to save your changes, and then select Yes.
4. Press Esc to return to the Internetworking Configuration menu.
5. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Configuring Probe Timeout

1. Load INETCFG and then select Protocols > TCP/IP > Dead Gateway Detection Configuration.
2. Select Probe Timeout and then enter a value in seconds.
3. Press Esc until you are prompted to save your changes, and then select Yes.
4. Press Esc to return to the Internetworking Configuration menu.
5. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Configuring Load Balancing

Before you configure load balancing check whether you have bound the desired NIC. If not, go to "Binding Protocols" on page 52 to do so.

1. Load INETCFG and select Protocols.
2. Select TCP/IP from the list of Protocols.
3. Select Load Balancing and enable it.

   This switch is for the complete system. You still have to configure load balancing on your network.
4. Select Load Balancing Configuration.
5. Configure Load Balancing Interval and Configure Individual Groups.
6. In Configure Individual Groups, select the network listed and enable load balancing.

Configuring Fault Tolerance

Before you configure fault tolerance check whether you have bound the desired NIC. If not, go to "Binding Protocols" on page 52 to do so.

1. Load INETCFG and select Protocols.
2. Select TCP/IP from the list of Protocols.
3. Select Fault Tolerance and enable it.

   This switch is for the complete system. You still have to configure fault tolerance on your network.

4. Select Fault Tolerance Configuration.
5. Configure Fault Detection Interval, Minimum Error Level and Configure Individual Groups.
6. In the Configure Individual Groups, select the network listed and enable fault tolerance.

Configuring Router Discovery

Both IP routers and end nodes can use the ICMP Router Discovery Protocol. Routers use it to advertise themselves as an IP router and to answer queries from end nodes. End nodes use it to locate an IP router on their network. Your system acts as a router when Packet Forwarding is enabled for IP and acts as an end node when Packet Forwarding is disabled for IP.

NOTE: For an end node to locate an IP router by this method, it must also support the ICMP Router Discovery Protocol.

To configure router discovery on an interface do the following:

1. Load INETCFG and then select Bindings > *an existing binding* > Expert TCP/IP Bind Options > Router Discovery Options.
2. Select the Status field > Enabled.
3. Select Destination Address.

   This is the method by which the IP router or end node sends router discovery packets.
4. Select one of the following options:
   - Broadcast—Sends the packets to all nodes on the network.
   - Router Discovery Multicast—Sends the packets to an IP multicast address used specifically for router discovery exchanges. The packets are received only by nodes that understand this multicast address.
5. Press Esc until you are prompted to save your changes, and then select Yes.
6. Press Esc to return to the Internetworking Configuration menu.
7. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Configuring Type of Service (TOS)

Using TOS, you can set the type of service for the IP data packets going out through an interface over a network.

Assign a value to set the TOS and the precedence bits of the IP header for outgoing packets. To set only TOS bits, use a value between 0 and 15. To set TOS and precedence bits, use a value between 0 and 127.

The valid range of values for TOS is from 0 to 127. Default value is 0, which indicates normal TOS. The four TOS bits are minimize delay (0x10), maximize throughput (0x08), maximize reliability (0x04), and minimize monetary cost (0x02).

This value is set only if TOS is enabled for this interface, otherwise, the TOS value set using SET options will be used. To see how to use SET options, see "Configuration Using SET Options" on page 53.

NOTE: The TOS value for outgoing IP datagrams can be set by an application using the WINSOCK API SetSockOpt. The value set by an API takes the highest preference followed by the value set using the method shown in "Assigning a TOS Value" on page 77, and then the value set using SET options.

Enabling TOS

1. Load INETCFG and then select Bindings > *an existing binding* > Expert TCP/IP Bind Options.
2. Select TOS, then select Enabled.
3. Press Esc until you are prompted to save your changes, and then select Yes.
4. Press Esc to return to the Internetworking Configuration menu.
5. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Assigning a TOS Value

1. Load INETCFG and then select Bindings > *an existing binding* > Expert TCP/IP Bind Options.
2. Select TOS Value and then enter a value.
3. Press Esc until you are prompted to save your changes, and then select Yes.

4. Press Esc to return to the Internetworking Configuration menu.
5. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Configuring ARP

IP routers and end nodes use ARP to determine the physical address of a node to which they want to send a packet. ARP is enabled by default. For one node to send a packet to another, the sending node must know the physical address of the destination node. The sending node, knowing only the destination IP address, first checks its ARP table for an entry that maps the destination IP address to the destination physical address. If the sending node finds the entry, it inserts the physical address into the packet and sends it. If the sending node does not find the entry in its ARP table, it broadcasts an ARP address request to the network. The destination node replies to the request with its own physical address, which the sending node uses to send the packet and adds to its ARP table for future use.

ARP Cache Update Timeout is the time period (in seconds) for an entry to be removed from the ARP table, if the entry has not been updated. This value can be set only for a primary interface, and if *ARP Timer* is enabled. ARP Timer is disabled by default. For best performance, set the ARP Cache Update timeout value to be greater than or equal to the value of ARP Cache Stale Timeout.

The valid range of values for ARP Cache Update Timeout is from 240 to 14400 seconds. Default value is 300 seconds.

ARP Cache Stale Timeout is the time period (in seconds) for an entry to be removed from the ARP table, if the entry has not been used for some time. If multihoming is enabled then we can set this value for a primary IP interface if they are grouped for load balancing and fault tolerance. In all other cases this can be set on all IP interfaces. ARP Timer is disabled by default. For best performance, set the ARP Cache Stale Timeout value to be lesser than or equal to the value of ARP Cache Update Timeout.

The valid range of values for ARP Cache Stale Timeout is from 240 to 14400 seconds. Default value is 300 seconds.

An IP router uses Proxy ARP when devices attached to one of its interfaces do not support IP subnetting and are unaware that they must go through the router to reach devices on other subnets of the same IP network. A router using Proxy ARP replies to ARP requests intended for devices on other subnets, but does

so only if the device is reachable through the router. To determine whether the device is reachable, the router examines its own routing table.

Proxy ARP is required on the parent network of a stub subnet. The parent network has an IP address range that includes the IP address range of the stub subnet. The router responds to ARP requests sent on the parent network on behalf of devices on the stub subnet.

When both the parent and stub subnet are bound to IP interfaces, the router can detect the parent/stub subnet and automatically enable Proxy ARP for the appropriate interfaces. Even if Proxy ARP is not required, and not automatically enabled, you can still force it to be enabled with the Force Proxy ARP parameter.

You must enable Force Proxy ARP on each LAN interface on which the router must reply to ARP requests for destinations it can reach. Force Proxy ARP is disabled on each interface by default.

This section contains the following topics:

Disabling ARP

1. Load INETCFG and then select Bindings > *an existing binding* > Expert TCP/IP Bind Options > ARP Options.
2. Select Use of ARP > Disabled.
3. Press Esc until you are prompted to save your changes, and then select Yes.
4. Press Esc to return to the Internetworking Configuration menu.
5. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Enabling Proxy ARP

1. Load INETCFG and then select Bindings > *an existing binding* > Expert TCP/IP Bind Options > ARP Options.
2. Select Force Proxy ARP > Enabled.
3. Press Esc until you are prompted to save your changes, and then select Yes.
4. Press Esc to return to the Internetworking Configuration menu.
5. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Enabling ARP Timer

1. Load INETCFG and then select Bindings > *an existing binding* > Expert TCP/IP Bind Options > ARP Options.
2. Select ARP Timer and then select Enabled.
3. Press Esc until you are prompted to save your changes, and then select Yes.
4. Press Esc to return to the Internetworking Configuration menu.
5. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Configuring ARP Cache Update Timeout

1. Load INETCFG and then select Bindings > *an existing binding* > Expert TCP/IP Bind Options > ARP Options.
2. Select ARP Cache Update Timeout and then enter a value in seconds.
3. Press Esc until you are prompted to save your changes, and then select Yes.
4. Press Esc to return to the Internetworking Configuration menu.
5. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Configuring ARP Cache Stale Timeout

1. Load INETCFG and then select Bindings > *an existing binding* > Expert TCP/IP Bind Options > ARP Options.
2. Select ARP Cache Stale Timeout, then enter a value in seconds.
3. Press Esc until you are prompted to save your changes, then select Yes.
4. Press Esc to return to the Internetworking Configuration menu.
5. If you want these changes to take effect immediately, select Reinitialize System and select Yes to activate your changes.

Configuring Directed Broadcast Forwarding

A *directed broadcast* is a broadcast intended for all nodes on a nonlocal network. For example, the broadcast address 255.255.255.255 reaches all nodes on a network; the directed broadcast address 128.1.255.255 is intended for all nodes whose network address is 128.1.0.0. A router not directly attached to 128.1.0.0 simply forwards the directed broadcast packet to the next hop. A router on network 128.1.0.0—if it has directed broadcast forwarding enabled—accepts and forwards the packet to all nodes whose network address is 128.1.0.0. Routers connecting subnets of 128.1.0.0 also accept and forward the packet to the nodes on their respective subnets.

IMPORTANT: For all nodes on network 128.1.0.0 to receive the directed broadcast, each router attached to network 128.1.0.0 must have Directed Broadcast Forwarding enabled.

Enabling Directed Broadcast Forwarding

1. Load INETCFG and then select Protocols > TCP/IP > Expert Configuration Options.
2. Select Directed Broadcast Forwarding > Enabled.
3. Press Esc until you are prompted to save your changes, and then select Yes.
4. Press Esc to return to the Internetworking Configuration menu.
5. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Configuring Source Route Packet Forwarding

Source route packets enable you to determine the route a packet takes to reach its destination. This feature is disabled by default.

1. Load INETCFG and then select Protocols > TCP/IP > Expert Configuration Options.
2. Select Forward Source Route Packets. Enabled to permit forwarding IP source route packets.
3. Press Esc until you are prompted to save your changes, and then select Yes.
4. Press Esc to return to the Internetworking Configuration menu.
5. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Configuring BOOTP Forwarding

BOOTP is a protocol that enables end nodes to receive their IP addresses from a BOOTP server at startup time. If your internetwork has a BOOTP or DHCP server, you can configure your IP router to accept and forward BOOTP or DHCP requests to that server.

1. Load INETCFG and then select Protocols > TCP/IP > Expert Configuration Options > BootP Forwarding Configuration.
2. Select BootP Server List and then press Ins.
3. Enter the IP address of the BOOTP or DHCP server at the prompt, or press Ins to display a list of symbolic hostnames and addresses from the SYS:\ETC\HOSTS file.

   The server address appears in the BOOTP Servers screen.
4. Press Esc.
5. Select BootP Packet Forwarding > Enabled.
6. If you want to record the activity of the BOOTP forwarder, select Log Operation and then select one of the following options:

   - Log to BootP Screen—Logs BOOTP activity to the BOOTP screen. This is a separate screen that you can select and monitor from the NetWare console. The information logged to this screen is not saved to a file.

- Log to File—Logs BOOTP activity to the SYS:\ETC\BOOTP.LOG file by default. To use a different file, type its full path name in the Log File field.

7. If you do not want to record the activity of the BOOTP forwarder, select Do Not Log.
8. Press Esc until you are prompted to save your changes, and then select Yes.
9. Press Esc to return to the Internetworking Configuration menu.
10. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Configuring EGP

The Exterior Gateway Protocol (EGP) is an exterior routing protocol that is supported by the TCP/IP software. Exterior routing protocols exchange information between different Autonomous Systems (ASs). The local EGP gets the information about its own AS from the local Interior Gateway Protocols (IGPs). Usually, exterior routing protocols are used only when different companies or commercial services are being connected.

The information EGP receives from the IGP must be explicitly configured. The exterior routing protocol shares only the information specified in the outgoing route filters. This is desirable because you generally want to limit the information exchanged between different ASs.

To configure EGP do the following:

1. Load INETCFG and then select Protocols > TCP/IP > Expert Configuration Options.
2. Select EGP > Enabled.
3. Select EGP Configuration and configure the following parameters:
   - Autonomous System—Enter the autonomous system number. It identifies the autonomous system to which the router belongs. The router establishes an EGP neighbor relationship with routers in other autonomous systems.
   - Maximum Neighbors to Acquire—Enter the maximum number of concurrent EGP neighbors with which this router can exchange EGP network reachability information.

- Neighbor List—Select this field to add, modify, or delete EGP neighbors. This router attempts to establish a relationship with the configured EGP neighbors to exchange network reachability information. Press Ins. Configure the following parameters:

  Neighbor's Address—Press Ins to display a list of symbolic hostnames from the SYS:\ETC\HOSTS file. Select a host here or enter the address.

  Neighbor's Autonomous System—Enter the number of the autonomous system to which this EGP neighbor belongs. The router is able to be a neighbor with the EGP peer only when the router and the EGP peer are in different autonomous systems.

4. Press Esc until you are prompted to save your changes, and then select Yes.
5. Press Esc to return to the Internetworking Configuration menu.
6. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Configuring Multiple Logical Interfaces

Novell TCP/IP allows you to bind more than one IP network to a LAN board. The networks can operate as separate logical interfaces. The ability to configure multiple logical interfaces simplifies the task of managing a growing network in the following ways:

- You can merge network when a there is a router failure.

  For a description, see “Merging Two Networks When the Connecting Router Fails” on page 85.

- You can move hosts from one IP network to another without losing connectivity.

  For a description, see “Reassigning IP Addresses” on page 86.

- You can add new nodes to a nearly full subnet.

  For a description, see “Adding New Nodes to a Full Subnet” on page 86.

To attach more than one IP network to a LAN board, bind IP to the board as many times as necessary; then supply a different IP address for each network.

Configuring multiple logical interfaces is different from multihoming, which enables you to bind multiple addresses belonging to the same IP network to

the same interface or different interfaces. To configure multihoming, see "Configuring a Secondary IP Address" on page 86.

Merging Two Networks When the Connecting Router Fails

Suppose a router that connects IP networks 130.81.0.0 and 167.10.0.0 fails. For simplicity, assume that the physical medium is Ethernet. If the router cannot be repaired quickly, you can temporarily fix the problem by completing the following steps:

1. Join the two networks into a single network segment using a barrel connector, a repeater, or other appropriate means.
2. Find a system operating Novell TCP/IP connected to the joined network.
3. Load INETCFG and select Protocols > TCP/IP.
4. Set IP Packet Forwarding to Enabled ("Router").
5. Press Esc until you are prompted to save your changes, and then select Yes.
6. Press Esc to return to the Internetworking Configuration menu.
7. Select Bindings and then bind IP to the joined network twice.
   - **7a** Select an existing binding to an interface connected to the joined network.
   - **7b** Set Local IP Address to an available host address on the first network. In this example, enter an available host address on the 130.81.0.0 network.
   - **7c** Press Esc and then save your change when prompted.
   - **7d** Press Ins to create a new binding and select the same interface connected to the joined network.
   - **7e** Set Local IP Address to an available host address on the second network. In this example, enter an available host address on the 167.10.0.0 network.
8. Press Esc until you are prompted to save your changes, and then select Yes.
9. Press Esc to return to the Internetworking Configuration menu.
10. If you want these changes to take effect immediately, select Reinitialize System > Yes.

Reassigning IP Addresses

Suppose you must change network number 89 to 130.57. If the system does not have multiple logical interfaces, you must change all IP addresses on network 89 at the same time or lose connectivity to any host that did not have its address changed.

With multiple logical interfaces, you can assign the new IP addresses gradually. Networks 89 and 130.57 can coexist on the same network segment until the transition is complete. The router interfaces, attached to *both* logical networks, forward packets for each network and route packets between the two.

Adding New Nodes to a Full Subnet

Suppose you want to add several new nodes to a subnet that has no more available IP addresses. Assume that the network has enough free connectors available to physically attach the nodes.

First, you assign a new subnet number to the cable so that both subnets share the cable. Then to add new nodes, you bind their IP address to the new logical subnet. The router whose interface is bound to both subnet addresses provides connectivity between the two subnets and to the rest of the internetwork.

Configuring a Secondary IP Address

A secondary IP address can be configured on the same interface that has the primary IP address.

When multiple interfaces exist, the secondary address is associated with the interface that is bound to an address that is on the same network. If there are more than one interface on the same network, then you can select the interface to add the secondary IP address. If the secondary address is not valid on any of the networks bound to existing interfaces, the address is rejected and an error message is produced.

To configure a secondary IP addresses, do the following:

1. Add a secondary IP address by entering the following at the command prompt:

   `add secondary IPAddress` ***`x.x.x.x noarp prompt`***

Noarp is used to add the secondary IP address as the non-ARPable one. If you don't use *noarp*, it will be added as ARPable. *Prompt* allows you to select from the available interfaces. If you don't use *prompt*, it will be added to the first bound interface of the same network.

2 Delete the secondary IP address by entering the following command:

```
del secondary IPAddress x.x.x.x
```

3 Display the secondary IP addresses by entering the following command:

```
display secondary IPAddress
```

6 Managing

This chapter describes the diagnostic utilities used to manage the Novell® TCP/IP software. These utilities enable you to manage, optimize, and troubleshoot the product and its connections. The following topics are discussed here:

Using the TCPCON Utility

TCPCON is an NLM™ utility that provides access to statistics and information about the status of various components of the TCP/IP protocol suite. TCPCON uses SNMP to access this information from any local or remote system on the network. TCPCON operates over TCP/IP and IPX™ networks.

To launch TCPCON, enter **LOAD TCPCON** at the system console prompt.

To monitor a remote system, select SNMP Access Configuration, change the Transport Protocol option to TCP/IP, and set the Host option to the IP address of the remote host you want to monitor. Press Esc to exit and save the options. If details from that remote host are displayed, there is a bidirectional route available.

You can use TCPCON to perform the following tasks:

- Monitor activity in the TCP/IP network segments of your internetwork
- Display configuration information and statistics about the following TCP/IP protocols—IP, ICMP, UDP, TCP, OSPF, and EGP
- Display the IP routes currently known to a TCP/IP node
- Display the network interfaces supported by a TCP/IP node
- Access the trap log maintained by SNMPLOG (for the local system only)
- Access TCP/IP information in any remote protocol stack supporting the TCP/IP Management Information Base (MIB)

HINT: TCPCON requires SNMP to be loaded on the remote host; otherwise, you receive an error message that the host is unavailable. Another cause of the Host unavailable message might be a routing error. To check for errors in the routing table, accept the default value of 127.0.0.1 in the Host option under SNMP Access Configuration. Select Routing Table to view the routing information table that the routing software has received from routing protocols (RIP and OSPF) or static routes. Compare this to the address topology of the network.

Viewing TCP/IP Configuration Information

To see how TCP/IP is configured, load TCPCON and select the following options:

- SNMP Access Configuration to view and change SNMP access configuration
- Protocol Information to view and change the run-time configuration of TCP/IP protocols
- IP Routing to view, change, and create IP routes
- Statistics to view detailed TCP/IP statistics
- Interfaces to view information about network interfaces
- Display Local Traps to view the local system SNMP trap log

Determining Whether a Remote TCP/IP Node Is Reachable

To determine whether a remote node is reachable, run an Echo test by doing the following:

1. Load PING
2. Specify the remote node address in the Host Name field.
3. Specify the number of seconds between each transmission in the Seconds to Pause between pings field.
4. Specify the packet size to be transmitted in the IP Packet Size to send in Bytes field.
5. Press Esc to begin transmitting.

If you receive an echo reply packet, the remote node is reachable.

Monitoring Error Counters

Error counters are monitored to make sure they are not increasing rapidly, because a rapid increase indicates a problem. For information about troubleshooting these problems, see “Troubleshooting” on page 93. You can monitor error counters for TCP/IP interfaces in the following ways:

- Use MONITOR to view counters such as Checksum Errors, Send and Receive Packet Errors, and interface-specific errors. To view these counters, load MONITOR and select LAN/WAN information > ***interface you want to view.***
- Use TCPCON to view the following TCP/IP counters:
  - IP Errors
  - IP Address Errors
  - Unknown Protocol Errors
  - Local Errors
  - Reassembly Failures Detected
  - Fragmentation Failures Detected

  To view these counters, load TCPCON and select Statistics > IP > More IP Statistics.

Monitoring TCP/IP Information

Monitoring TCP/IP information can give you a clear view of the status of your TCP/IP network and whether the router is configured properly to run efficiently in the network. This information can also be helpful in troubleshooting and optimizing of the network.

Checking the TCP/IP Routing Table

To check the TCP/IP routing table and information associated with each route, load TCPCON and select IP Routing Table > Proceed > *entry you want to view*.

The IP Routing Table window shows you all known TCP/IP destination networks and the following information about each item:

- IP address of the destination
- IP address of the next hop router
- Type of the route (direct, remote)
- Primary cost for the route
- Interface used to reach a route

The IP Route Information window expands on this by showing information about the mask used, the routing protocol through which the destination was learned, and the age of the route.

Monitoring the Configured TCP/IP Protocols

You can view, and sometimes change, the configuration of TCP/IP protocols configured for use in your router. You can reach this information by loading TCPCON and selecting Protocol Information. You can configure and view statistics and other information for the following protocols:

- EGP
- ICMP
- IP
- OSPF
- TCP
- UDP

For additional information about each protocol, press F1 to access online help.

7 Troubleshooting

This chapter contains IP troubleshooting information that is divided into three categories:

If a problem that is general in nature occurs, the procedure described in Troubleshooting Checkpoints will help you isolate and resolve the problem.

Troubleshooting Tools

TCPCON is an NLM™ utility that provides access to statistics and information about the status of various components of the TCP/IP protocol suite. It uses SNMP to access this information from any local or remote system on the network. TCPCON operates over TCP/IP networks. Use TCPCON to monitor a remote system.

You can use TCPCON to perform the following tasks:

- Monitor activity in the TCP/IP network segments of your internetwork
- Display configuration information and statistics about the following TCP/IP protocols: IP, ICMP, UDP, TCP, OSPF, and EGP
- Display the IP routes currently known to a TCP/IP node
- Display the network interfaces supported by a TCP/IP node
- Access TCP/IP information in any remote protocol stack supporting the TCP/IP Management Information Base (MIB)

Troubleshooting Checkpoints

To isolate and resolve TCP/IP problems, do the following:

1. To verify that IP is bound to the desired interfaces with the correct addresses and masks for your internetwork. Load TCPCON Select Protocols > IP > IP Addresses.

   Use INETCFG to make any required corrections.

2. To check the routing table for routes to the required network. Load TCPCON and select IP Routing Table > Proceed, and then press Enter

   If routes are missing, verify that the required routing protocols have been enabled and bound to the correct interfaces in INETCFG. Also verify that the routing protocol in use on an interface is correctly configured on other routers that are accessible through that interface.

3. To verify that static routing is configured if other third-party routers that do not use RIP or OSPF are connected on the network. Load INETCFG and select Protocols > TCP/IP > LAN Static Routing

4. To verify that the IP Packet Forwarding statistic is set to Enabled. Load INETCFG and select Protocols > TCP/IP > IP Packet Forwarding

   Use INETCFG to make any required corrections and then reinitialize the system.

5. Use PING or TPING to test connectivity.

   Perform Step 1 through Step 4 on any routers that cannot be reached. Start with the router that is closest to the local node.

6. Verify that all client software has the Default Router parameter configured to match the IP address of the network board inside the router that is connected to the local segment.

7. Load TCPCON for the following IP statistics:

   - Local errors (memory error)
   - IP errors (unexpected protocol errors)

     Check the configuration of other IP nodes on the network. Reduce IP traffic or use a network analyzer to identify the source of invalid packets.

   - IP address errors (misdirected packets)

     Check the Address Translation tables on other IP nodes to determine the source of the errors.

- Unknown protocol errors (unsupported IP clients)

  Load the required applications.

- No route found (router failure)

  Check the configuration of the routing protocols.

8. Load TCPCON for the following ICMP statistics:

   - Destination unreachable (network failure)

     Use a network analyzer to determine the unreachable destination. Check that the routers on the path to the destination advertise the route.

   - Time exceeded (network failure)

     Reduce the excessive delays by reducing the size of the internetwork or increasing the speed of WAN links.

   - Redirects (router failure)

     Check that all routers on the network are properly configured and advertising routes. Verify that the correct Default Router is configured on the clients.

9. To verify that all configuration options are set correctly. Load INETCFG and select View Configuration

Common Problems

This section discusses the following common problems and their potential solutions:

- “Losing INETCFG Configuration Information upon Rebooting” on page 101
- “Loss of Secondary IP Address upon Deleting Any Binding” on page 101

LAN Connectivity Problems

- The router does not forward IP packets

  Verify that the IP Packet Forwarding statistic is set to Router Enabled in TCPCON (Protocols > IP). If routing is not enabled, enable IP Packet Forwarding under Protocols in INETCFG, and then issue the REINITIALIZE SYSTEM command.

- A TCP/IP host cannot reach the router on the local network
  - Verify that the network portion of the IP address and the subnet mask are the same on the router and the host.
  - Verify that the router and host use the same frame type.
  - Use PING from the router to verify connectivity to the TCP/IP host and verify that the IP Address Translation table has an entry for the host.

    If there is no entry, use MONITOR to check the status of the LAN driver.
  - Use PING from the router to verify connectivity to the TCP/IP host and check for Echo Requests in TCPCON (select Statistics > ICMP).

    If the value of the Echo Requests statistic is not incrementing, check the IP statistics for errors and perform Step 7 on page 94 in Troubleshooting Checkpoints.
  - Use PING from the router to verify connectivity to the TCP/IP host and check for Echo Replies in TCPCON (select Statistics > ICMP).

    If the value of the Echo Replies statistic is not incrementing, verify that IP is bound to the host's interface with the correct address and mask. Also, verify that the interface driver is loaded with the correct frame type. If required, check the IP statistics for errors and perform Step 7 on page 94 in Troubleshooting Checkpoints.

- A TCP/IP host cannot reach a remote host
  - Verify that the IP address and mask are proper.
  - Verify that the local TCP/IP host has the local router listed as the default router.
  - Verify that each router has a routing protocol enabled and that it has not been disabled on the interface.
  - Starting at the local router, verify that each router has a route to the remote host's network.
  - Verify that there are no filters capable of blocking IP traffic configured on any routers along the path.
  - Verify that the remote host has a route to the local host's network.
  - Using PING, verify that the remote host can reach each router on the path to the local TCP/IP host.
  - Starting at the router closest to the remote host, verify that each router has a route to the local TCP/IP host's network.
- The router cannot initiate IP traffic to a remote router through a LAN interface
  - Verify that IP is bound to the right interface with the correct address and mask.
  - Check whether the interface driver is loaded with the correct frame type.
  - Check whether a route exists to the network on which the destination router resides. This can be done through the IP Routing Table window of TCPCON. If the destination router is accessible, also verify that it has a route to the source router's network.
- A TCP/IP host cannot reach another host when Fault Tolerance is disabled and the NIC that was handling the data transfer has gone down
  - Make sure the NIC is grouped for load balancing and fault tolerance.
  - Make sure that fault tolerance is enabled for the group.

To verify that NIC is grouped, do the following:

1. Load INETCFG and select Bindings > TCP/IP.
2. Select Configure TCP/IP Bind Option.
3. Make sure that the Group interface for LBFT is set to Yes.

To check that fault tolerance is enabled, do the following:

1. Load INETCFG and select Protocols > TCP/IP.
2. In the screen that appears, fault tolerance should be enabled.

   This could be for the complete system. You still need to verify whether Fault Tolerance is enabled for the particular Net Group.
3. Select Fault Tolerance Configuration > Configure Individual Group.
4. Select the particular Net Group and check whether fault tolerance is enabled for it. If not, set it to Yes.

Router Cannot Ping a Remote Router or the Internet

Load TCPCON and verify select IP Routing Table that there is a destination that is specified as the default route. If there is no default route, you must configure it. Load INETCFG to permanently configure the default route.

1. Load INETCFG and select Bindings > TCP/IP.
2. Select the Interface Group for your WAN card.
3. Select WAN Call Destination.
4. Select WAN Call Destination press Ins and then select the WAN card defined earlier.
5. Select Static Routing Table option and press Ins.

   The Static Routing entry sets up the default route that points to the Internet Service Provider (ISP).
6. Select Route to Network or Host > Default Route.
7. Press Esc to save your changes and exit the menus.
8. For the changes to take effect, reinitialize the system.

Routing Table Maintenance Problems

- Routes are not exchanged on a LAN.
  - Use INETCFG to verify that the IP Packet Forwarding option is enabled.
  - Use INETCFG to verify that a routing protocol has been enabled.
  - Use INETCFG to verify that the routing protocol has not been disabled on an interface.

- Use TCPCON to examine the routing table and determine which routes are missing.
- Check TCPCON for IP errors.
- Check TCPCON for ICMP errors.
- If you are using RIP, then in INETCFG under Bindings, verify that the RIP Mode option is not set to Send Only or Receive Only.
- If the RIP Version option is set to RIPII, verify that the other routers also support RIP II.
- Verify that no route filters are configured that would block route information packets for that interface.
- If you are using OSPF, verify that the following conditions have been met:

  Routers in the area have the same Authentication Type configured.

  All routers on the same network have the same Authentication Password configured for the interface to the network.

  All routers on the same network have the same Hello Intervals configured for the interface to the network.

  The state of each neighbor is either two-way or full in TCPCON (select Protocol Information > OSPF > Neighbors). If it is not, one of the two conditions described next will occur. Refer to the next two paragraphs for an explanation of the corrective actions required.

  In TCPCON, there is a router link state advertisement for each router in your area (select Protocol Information > OSPF > Link State Advertisements). If these advertisements are not present, verify that the missing router is active and the correct area ID is configured for the network interface.

  In TCPCON, the number of link state advertisements, Area Boundary Routers, and Autonomous System Boundary Routers are the same for each router in your area (select Protocol Information > OSPF > Areas). Verify that the problem routers are active. Bring down any router whose routing database is not synchronized with the databases of its routing neighbors. If the problem persists, reduce the size of your network or add more memory to the router.

- Routes are not exchanged on a LAN.

  Verify that the broadcast address is correct.

- RIP routes are not accessible to hosts on OSPF networks.
  - Check the status of the Autonomous System Boundary Router statistic in TCPCON (select Protocol Information > OSPF).
  - Verify that no filters are configured that would block access to the network.

IP Address Duplication across Machines

- When you are trying to bind an IP address, you get an error message stating a conflict for the IP address.
  - Set **`allow ip address duplicates`** command off under SET parameters.

Server Not Responding under Heavy Stress Conditions

Increase the following using the SET command options:

Maximum Packet Receive Buffers (page 56)

Minimum Packet Receive Buffers (page 56)

TCP IP Maximum Small ECBs (page 58)

Load Not Balanced across NICs although LB is Enabled in INETCFG

Load balancing might have been enabled only for the system and not for the particular group. Check whether you have grouped multiple NICs and enabled load balancing for them.

To check that load balancing is enabled at group level, do the following:

1. Load INETCFG and select Protocols > TCP/IP.

   In the screen that appears load balancing should enabled. This is for the complete system. You still need to verify whether load balancing is enabled for the particular Net Group.
2. Select Load Balancing Configuration > Configure Individual Group.
3. Select the particular Net Group and check whether load balancing is enabled for it. If not, set it to Yes.

Network Traffic Is Not Balanced across NICs

- If the application is binding to the local host (0.0.0.0), the data is always sent through the Primary. So the Primary should be inside the LBFT group for the load to be evenly balanced.

  Check whether the interface designated as Primary is within the group or not. If not, either group the Primary interface or make one of the group members Primary.

- If the application is not bound to the local host, then the data is always sent through the host where the application is bound. To load balance in this case, make sure that this host is inside the LBFT Group.

To check whether a particular binding is part of the LBFT Group, do the following:

1. Load INETCFG and select Bindings > TCP/IP.
2. In the screen that appears, select the Configure TCP/IP Bind option.
3. Make sure that the group interface for LBFT is set to Yes.

Losing INETCFG Configuration Information upon Rebooting

This could happen if the server abended while being configured. This corrupts the file SYS:\ETC\TCPIP.CFG. Delete this file and copy a backup of the previous configuration.

Loss of Secondary IP Address upon Deleting Any Binding

This could happen if more than one interface are using the same driver. If this is the case, never delete any of the bindings. Always disable them if you don't want to use them.

A Planning

This appendix explains what decisions must be made before you can configure TCP/IP beyond its most basic configuration.

Configuration Decisions

How you configure TCP/IP beyond the most basic configuration depends on the following decisions:

- **Whether a multiprocessor server can use this version of TCP/IP**

  The TCP/IP stack distributes the connection across all processors uniformly resulting in packet processing on different processors in parallel. The TCP/IP stack has been multiprocessor (MP) enabled for processing TCP and UDP packets.

- **Whether to use the computer as a router or an end node (that is, a host)**

  The IP Packet Forwarding parameter, which controls IP packet routing, is enabled by default. This parameter permits your computer to operate as an IP router. When you want your computer to operate as an end node only, disable this parameter.

- **Whether to use Routing Information Protocol (RIP), Open Shortest Path First (OSPF), or a mixed RIP-OSPF environment**

  RIP and OSPF are IP routing protocols. If you already have IP routers in your network environment, use the same routing protocol they use. If your network currently has no other routers, use OSPF.

  To configure your router as a RIP router, see "Configuring RIP" on page 60.

To configure your router as an OSPF router, see “Configuring OSPF” on page 63.

To configure a mixed RIP-OSPF environment, refer to both of the preceding procedures.

- **Whether to use static routes on a router**

  Static routes are useful for reducing routing traffic, providing security, accessing isolated networks, and operating as backup routes on routers. Static routes are required for on-demand connections.

  To configure static routes on a router, see “Configuring Static Routes for LANs” on page 69.

- **Whether to filter routes or various TCP/IP packets**

  Enable filters when you want to do either of the following:

  - Control access to any services, such as File Transfer Protocol (FTP), on your network
  - Reduce the bandwidth consumed by routing traffic

  To configure TCP/IP filters, you must enable the Filtering Support parameter in INETCFG and then load the Filter Configuration utility (FILTCFG).

- **Whether to configure router discovery**

  Router discovery enables end nodes to find an IP router on their network. If your computer is operating as a router, it can advertise itself periodically as a router. If your computer is operating as an end node, it can send queries to locate a router.

  To configure router discovery, see “Configuring Router Discovery” on page 76.

- **Whether to disable Address Resolution Protocol (ARP) or enable Proxy ARP**

  ARP is a LAN protocol that maps Internet addresses to physical addresses. IP routers and end nodes use ARP to determine a destination node's physical address.

  An IP router using Proxy ARP replies to ARP requests it receives through an interface on behalf of an end node on a network attached to another interface.

  To change the default settings of ARP and Proxy ARP features, see “Configuring ARP” on page 78.

- **Whether to enable the router to forward directed broadcasts**

  A *directed broadcast* is a broadcast intended only for a subnet rather than all nodes on the network.

  To enable directed broadcast forwarding, see "Configuring Directed Broadcast Forwarding" on page 81.

- **Whether to configure the router or end node as a BOOTP forwarder**

  The BOOTP protocol enables end nodes to obtain an IP address at start-up time. If there is a BOOTP or Dynamic Host Configuration Protocol (DHCP) server on your internetwork, any IP routers that are configured to act as a BOOTP forwarder accept and forward BOOTP or DHCP requests to the server. The BOOTP or DHCP server then assigns an IP address to the end station.

  To configure BOOTP forwarding, see "Configuring BOOTP Forwarding" on page 82.

- **Whether to configure multiple logical interfaces on a single board**

  Using multiple logical interfaces enables you to bind more than one IP network to a LAN board. Each binding operates as a separate logical interface.

  To configure multiple logical interfaces on a board, see "Configuring Multiple Logical Interfaces" on page 84.

- **Whether to use multihoming**

  Multihoming enables an interface to be bound to multiple IP addresses on the same network. Multihoming can be used for all IP networks bound to a router, whether the networks are bound to on the same interface or different interfaces.

  The most common use of multiple addresses on the same network is to enable a Web server to operate as though it is several Web servers. In this application, each secondary IP address is used by a different virtual host on the same Web server. The Domain Name System (DNS) can be used to access these virtual hosts using unique hostnames.

  Multihoming is also commonly used with network address translation (NAT), the proxy server, and the virtual private network (VPN).

  To configure multihoming, see "Configuring a Secondary IP Address" on page 86.

B TCP/IP Database Files

This appendix describes how the database files of the TCP/IP stack should be configured.

Configuring Database Files

TCP/IP uses four database files in the SYS:\ETC\directory to convert internal data, such as IP addresses, into more identifiable and workable names.

- HOSTS File (page 108) maps hostnames to IP addresses
- NETWORKS File (page 109) maps network names to network addresses
- PROTOCOL File (page 110) maps protocol names to IP protocol numbers
- SERVICES File (page 111) maps service names to TCP and UDP ports

The user interface for TCPCON and other NLM files uses these database files. To inform TCP/IP of names and addresses of local nodes and networks, you must add that information to these files. The files are cached in memory so that disk access is avoided during lookup. Because of this, TCP/IP takes up more memory. If this is not desirable, keep the size of the database small or simply delete the files.

TCP/IP finds the following four database files in the SYS:\ETC\directory:

These files are described in the sections following this discussion.

If you are configuring TCP/IP for the first time, we recommend that you start by copying the sample database files from SYS:\ETC\SAMPLES to SYS:\ETC. This provides you with some examples to refer to as you add your own entries, and also provides TCP/IP with the PROTOCOL and SERVICES files.

You can modify these files with a standard text editor from any NetWare® client, or you can use EDIT.NLM from the NetWare system console. The following sections describe the formats of the files, which are compatible with the same files on standard 4.3BSD UNIX* systems. The examples in the sample files can also help you create your own entries.

The files have the same names and format as the files on UnixWare* systems and other UNIX systems. You can use FTP to transfer the files from a UNIX host.

Each database file describes a table. Each line of the file describes a separate table entry. Blank lines and comments are ignored. Comments begin with a pound sign (#) anywhere in a line and include the pound sign and any characters following it on the same line.

IMPORTANT: Do not use the sample addresses provided in the database files if you are connected to the Internet; these addresses are for example only.

HOSTS File

The SYS:\ETC\HOSTS file contains information about the known hosts on the IP internetwork. Typically, it is centrally administered and distributed to all local hosts. Its format, as shown in Figure 15, is identical to /etc/hosts on UNIX systems. Each entry provides information about a single host. An entry cannot extend beyond one line.

Figure 15 Sample HOSTS File

```
#
# Mappings of host names and host aliases to IP addresses
#
127.0.0.1       loopback lb localhost  # loopback address
#
# examples from a fictitious network
#
129.47.4.2      ta tahiti ta.some.com loghost
129.47.6.40     osd-frog frog
129.47.6.144    sj-in5 in5
192.67.172.71   sj-in1 in1
```

The HOSTS file entry has the following format:

IP_address host_name [*alias* [...]]

The *IP_address* is a 4-byte address in standard dotted decimal notation. Each byte is a decimal, hexadecimal, or octal value and is separated by a period. Hexadecimal numbers must start with the character pair 0x or 0X; octal numbers must start with 0.

The *host_name* is the name of the system associated with this IP address. The name cannot contain a space, tab, pound sign (#), or end-of-line character. Each hostname must be unique.

The *alias* is another name for the same system. Typically, this is a shorter name. A single host can have from 1 to 10 aliases. For example, the host sales could have the following address and aliases:

```
129.0.9.5 sales sa saleshost
```

The sample file SYS:\ETC\SAMPLES\HOSTS is included with the TCP/IP software. When you are configuring TCP/IP for the first time, copy the sample HOSTS file from SYS:\ETC\SAMPLES to SYS:\ETC. You then edit the SYS:\ETC\HOSTS file. You can change your configuration at any time by editing your existing SYS:\ETC\HOSTS file.

NETWORKS File

The SYS:\ETC\NETWORKS file contains information about the networks in your internetwork. Each entry provides information about one network. An entry cannot extend beyond one line. Figure 16 shows a sample NETWORKS file.

Figure 16 Sample NETWORKS File

```
#
# Network numbers
#
loopback    127      # fictitious internal loopback network
somenet     129.47   # fictitious network number
#
# Internet networks
#
arpane      10 arpa # historical network
milnet      26      # military network
```

The NETWORKS file entry has the following format:

```
network_name network_number [/network_mask] [alias [...]]
```

The *network_name* is the name of the network associated with this network number. The name cannot contain a space, tab, pound sign (#), or end-of-line character. The network name must be unique.

The *network_number* is the number of the network. Hexadecimal numbers must start with the character pair 0x or 0X. The *network_number* can be specified with or without trailing zeros. For example, the addresses 130.57 and 130.57.0.0 denote the same IP network.

The *network_mask* is the subnet mask of the network. Like IP addresses, it can be specified in octal, decimal, or hexadecimal notation. This field is optional. If not specified, the subnet mask is deduced from existing routing table entries.

The *alias* is another name for the same network; you can specify up to 10 aliases for a network.

The sample file SYS:\ETC\SAMPLES\NETWORKS is included with the TCP/IP software. When you are configuring TCP/IP for the first time, copy the sample NETWORKS file from SYS:\ETC\SAMPLES to SYS:\ETC. Then edit the SYS:\ETC\NETWORKS file. You can change your configuration at any time by editing your existing SYS:\ETC\NETWORKS file.

PROTOCOL File

The SYS:\ETC\PROTOCOL file, as shown in Figure 17, contains information about the known protocols used on the internetwork. Each line provides information about one protocol. An entry cannot extend beyond one line.

NOTE: The PROTOCOL file is called PROTOCOLS on UNIX systems. The name is shortened to PROTOCOL because of the DOS eight-character limit.

Figure 17 Sample PROTOCOL File

```
#
# Internet (IP) protocols
#
ip     0 IP    # internet protocol, pseudo protocol number
icmp   1 ICMP  # internet control message protocol
igmp   2 IGMP  # internet group multicast protocol
ggp    3 GGP   # gateway-gateway protocol
tcp    6 TCP   # transmission control protocol
pup    12PUP   # PARC universal packet protocol
udp    17UDP   # user datagram protocol
```

The PROTOCOL file entry has the following format:

```
protocol_name protocol_number [alias [...]]
```

The *protocol_name* is the name of the Internet protocol associated with this protocol number. The name cannot contain a space, tab, pound sign (#), or end-of-line character.

The *protocol_number* is the number of the Internet protocol.

The *alias* is an alternate name for the protocol.

The sample file SYS:\ETC\SAMPLES\PROTOCOL is included with the TCP/IP software. When you are configuring TCP/IP for the first time, copy the sample PROTOCOL file from SYS:\ETC\SAMPLES to SYS:\ETC. You can then edit the SYS:\ETC\PROTOCOL file. You can change your configuration at any time by editing your existing SYS:\ETC\PROTOCOL file.

SERVICES File

The SYS:\ETC\SERVICES file, as shown in Figure 18, contains information about the known services used on the IP internetwork. Each entry provides information about one service. An entry cannot extend beyond one line.

Figure 18 Sample SERVICES File

```
#
# Network services
#
echo      7/udp
echo      7/tcp
discard   9/udp     sink null
discard   9/tcp     sink null
tftp      69/udp
login     513/tcp
shell     514/tcp   cmd
```

The SERVICES file entry has the following format:

```
service_name port_number /protocol_name [alias [...]]
```

The *service_name* is the name of the service associated with this port number and protocol name. The name cannot contain a space, tab, pound sign (#), or

end-of-line character. These are generally Application-layer, Presentation-layer, or Session-layer services, such as TFTP, FTP, SMTP, and TELNET.

The *port_number* is the number of the Internet port used by the service.

The *protocol_name* is the protocol with which the service is associated. This is generally a Transport- or Network-layer protocol, such as TCP or UDP. You must put a slash between the port number and the protocol name (for example, SMTP 25/TCP MAIL).

The *alias* is an alternate name for the service.

The sample file SYS:\ETC\SAMPLES\SERVICES is included with the TCP/IP software. When you are configuring TCP/IP for the first time, you should copy the sample SERVICES file from SYS:\ETC\SAMPLES to SYS:\ETC. You can then edit the SYS:\ETC\SERVICES file. You can change your configuration at any time by editing your existing SYS:\ETC\SERVICES file.