O N E

TCP/IP Overview

TCP/IP for Windows 2000

Transmission Control Protocol/Internet Protocol (TCP/IP) for Windows 2000 is:

- Networking software based on industry-standard networking protocols.
- A scalable, enterprise networking protocol that supports the connection of your Windows-based computer to both LAN and WAN environments.
- Core technologies and utilities for enabling your Windows-based computer to connect to and share information with dissimilar systems.
- A foundation for gaining access to global Internet services, such as the World Wide Web and File Transfer Protocol (FTP) servers.
- A robust, scalable, cross-platform, client/server framework.

Because most modern operating systems support the TCP/IP protocol suite, you can use TCP/IP to share information with a wide variety of systems and resources, as shown in the Figure 1-1.

TCP/IP for Windows 2000 provides basic TCP/IP utilities that enable a computer running Windows 2000 to connect and share information with other Microsoft and non-Microsoft systems, including:

- Internet hosts
- Apple Macintosh OS systems
- IBM mainframes
- UNIX systems
- Open VMS systems

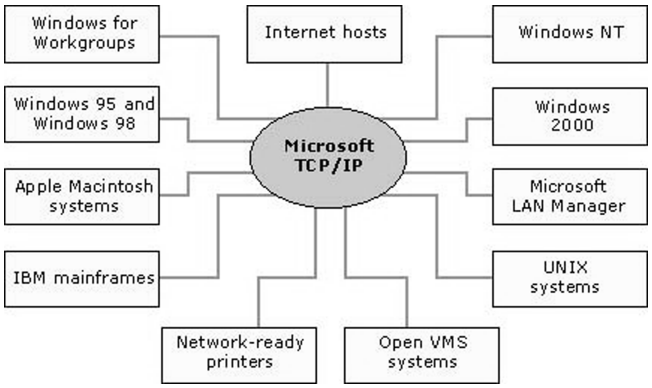

FIGURE 1–1 *How TCP/IP supports open system use*

- Microsoft Windows NT and Windows 2000
- Microsoft Windows 98
- Microsoft Windows 95
- Microsoft Windows for Workgroups
- Microsoft LAN Manager
- Network-ready printers, such as HP LaserJet series printers that use HP JetDirect cards

TCP/IP Background

TCP/IP is an industry-standard suite of protocols designed for large-scale internetworks that span LAN and WAN environments.

As the timeline in Figure 1-2 shows, the origins of TCP/IP began in 1969, when the U.S. Department of Defense (DoD) commissioned the Advanced Research Projects Agency Network (ARPANET).

The ARPANET was the result of a resource-sharing experiment. The purpose was to provide high-speed network communication links between various supercomputers located at various regional sites within the United States.

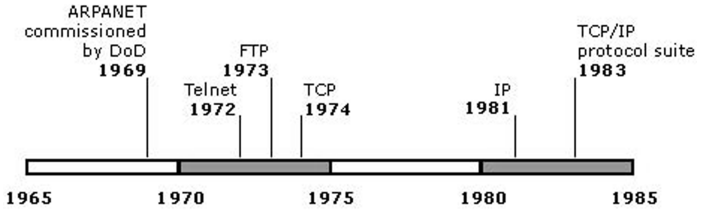

FIGURE 1–2 *Timeline for TCP/IP*

Early protocols such as Telnet (for virtual terminal emulation) and the FTP were first developed to specify basic utilities needed for sharing information across the ARPANET. As the ARPANET grew in size and scope, two other important protocols appeared:

- In 1974, the Transmission Control Protocol (TCP) was introduced as a draft specification that described how to build a reliable, host-to-host data transfer service over a network.
- In 1981, the Internet Protocol (IP) was introduced in draft form and described how to implement an addressing standard and route packets between interconnected networks.

On January 1, 1983, ARPANET began to require standard use of the TCP and IP protocols for all network traffic and essential communication. From this date forward, ARPANET started to become more widely known as the *Internet,* and its required protocols started to become more widely known as the *TCP/IP protocol suite*.

The TCP/IP protocol suite is implemented in a variety of TCP/IP software offerings available for use with many computing platforms. Today, TCP/IP software remains widely in use on the Internet and is used often for building large, routed, private internetworks.

The TCP/IP Model

TCP/IP is based on a four-layer reference model. All protocols that belong to the TCP/IP protocol suite are located in the top three layers of this model.

As shown in Figure 1-3, each layer of the TCP/IP model corresponds to one or more layers of the seven-layer Open Systems Interconnection (OSI) reference model proposed by the International Standards Organization (ISO).

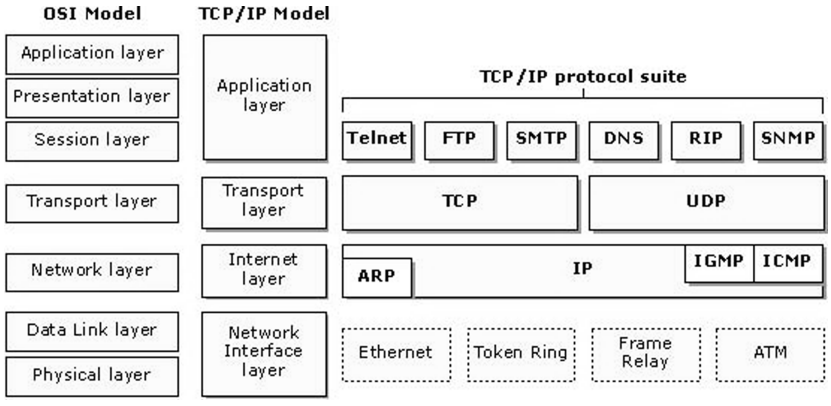

FIGURE 1–3 *OSI layer model and TCP/IP model*

The types of services performed and protocols used at each layer within the TCP/IP model are described in more detail in Table 1.1.

TABLE 1.1 *Services and Protocols in the TCP/IP Model*

Layer	Description	Protocols
Application	Defines TCP/IP application protocols and how host programs interface with transport layer services to use the network.	HTTP, Telnet, FTP, TFTP, SNMP, DNS, SMTP, X Windows, other application protocols
Transport	Provides communication session management between host computers. Defines the level of service and status of the connection used when transporting data.	TCP, UDP, RTP
Internet	Packages data into IP datagrams, which contain source and destination address information that is used to forward the datagrams between hosts and across networks. Performs routing of IP datagrams.	IP, ICMP, ARP, RARP
Network interface	Specifies details of how data is physically sent through the network, including how bits are electrically signaled by hardware devices that interface directly with a network medium, such as coaxial cable, optical fiber, or twisted-pair copper wire.	Ethernet, Token Ring, FDDI, X.25, Frame Relay, RS-232, v.35

For more information about Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), Internet Group Management Protocol (IGMP), User Datagram Protocol (UDP), IP, and TCP, see "Understanding TCP/IP" in this chapter.

Note The OSI Reference Model (OSI RM) is not specific to TCP/IP. Beginning in the late 1970s, the ISO developed it as a generic framework describing all aspects of communication protocols. The OSI RM is a widely known and accepted reference model in the data communications field and is used here only for comparison purposes.

TCP/IP Utilities

Windows 2000 includes three types of TCP/IP-based utilities:

- Connectivity utilities that you can use to interact with and use resources on a variety of Microsoft and non-Microsoft hosts, such as UNIX systems.
- Diagnostic utilities that you can use to detect and resolve networking problems.
- TCP/IP server software that provides printing and publishing services to TCP/IP-based clients on Windows 2000.

Tables 1.2, 1.3, and 1.4 briefly describe both familiar and some new TCP/IP-based utilities in Windows 2000. For more information on using these utilities and their syntax, refer to Appendix E, "TCP/IP Utilities."

TABLE 1.2 *Connectivity Utilities*

Utility	Description
FTP	Transfers files of any size between Windows 2000 and any computer running FTP server software.
LPR	Sends print jobs to remote UNIX printers managed by Line Printer Daemon (LPD) print server software.
RCP	Copies files between Windows 2000 and computers running Remote Copy Protocol (RCP) server software.
REXEC	Executes processes on remote computers.
RSH	Runs commands on a computer running Remote Shell (RSH) server software.
TELNET	Uses terminal-based login to remotely access network devices that are running Telnet server software.
TFTP	Transfers small files between Windows 2000 and computers running Trivial File Transfer Protocol (TFTP) server software.

TABLE 1.3 *Diagnostic Utilities*

Utility	Description
ARP	Displays and modifies the ARP cache. This cache is a local table used by Windows 2000 to resolve IP addresses to media access control addresses used on the local network.
Hostname	Returns the host name of the local computer.
IPCONFIG	Displays the current TCP/IP configuration. Also used to manually release and renew TCP/IP configurations assigned by a DHCP server.
LPQ	Obtains print queue status information from computers running LPD print server software.
NBSTAT	Displays the local NetBIOS name table, a table of NetBIOS names registered by local applications, and the NetBIOS name cache, a local cache listing of NetBIOS computer names that have been resolved to IP addresses.
Netstat	Displays TCP/IP protocol session information.
NSLOOKUP	Checks records, domain host aliases, domain host services, and operating system information by querying Domain Name System (DNS) servers.
PING	Verifies configurations and tests IP connectivity.
ROUTE	Displays or modifies the local routing table.
TRACERT	Traces the route a packet takes to a destination.
PATHPING	Traces the route a packet takes to a destination and displays information on packet losses for each router in the path. Pathping can also be used to troubleshoot Quality of Service (QoS) connectivity.

TABLE 1.4 *Server-based Software*

Server software	Description
TCP/IP Printing service	The TCP/IP Printing service offers an LPD printing service for computers running Windows 2000. This service allows UNIX computers to send print jobs to computers running Windows 2000 by using the Line Printer Remote (LPR) utility for printing.
Internet Information Services	Internet Information Services (IIS) offers Web, Gopher, and FTP server software for providing TCP/IP-based publishing services. IIS is provided with Windows 2000 Server to offer an enterprise-ready Web server to use with an unlimited number of simultaneous connections.
Peer Web Services	Peer Web Services is supplied with Windows 2000 Professional to provide Web publishing services similar to those offered by IIS for serving fewer than 10 simultaneous connections.

New TCP/IP Features for Windows 2000

TCP/IP for Windows 2000 has been updated to include several features that simplify configuration on a single subnet and optimize TCP performance in a wider variety of network environments.

These new features include support for:

- Automatic private address configuration (pre-standard).
- Large TCP windows (RFC 1323).
- TCP selective acknowledgments ("sack") (RFC 2018).
- Better TCP round trip time (RTT) estimation (RFC 1323).
- ICMP Router Discovery Protocol (RFC 1256).
- DNS caching.
- Disabling NetBIOS over TCP/IP.

Automatic Private Address Configuration

Automatic Private IP Addressing (APIPA) is used to allocate addresses within subnetworks that either do not have a local DHCP server or do not have a routed path to a DHCP server on another subnetwork.

By default, a computer running Windows 2000 first tries to contact a DHCP server to obtain an IP address and related configuration information for each installed network connection:

- If a DHCP server is reached and the leased configuration is successful, TCP/IP configuration is completed.
- If a DHCP server is not reached, the computer instead uses APIPA to automatically configure TCP/IP. When you use APIPA, Windows 2000

determines an address in the reserved IP address range from 169.254.1.1 through 169.254.254.254. This address is used as a temporary IP address configuration until a DHCP server is located. The subnet mask is set to 255.255.0.0.

The APIPA range of IP addresses (169.254.0.0/16) has been set aside for use in IP address autoconfiguration by the Internet Assigned Numbers Authority (IANA; *www.iana.org/*). Any IP addresses within this range are not usable on the Internet.

APIPA eliminates IP address configuration for single-network small office or home office networks that are not connected to the Internet.

Large TCP Windows

Window size reflects the maximum number of packets that can be sent without waiting for positive acknowledgment. Large TCP windows improve TCP/IP performance by allowing large amounts of data to be in transit between the sender and receiver. In typical TCP-based communication, the maximum window size is usually fixed at the onset of connection and is often statically limited to 64 kilobytes (65,536 bytes).

With large window support, the TCP connection endpoints can dynamically recalculate and optimize the actual window size by using a TCP option as needed during longer sessions. With this option, more data packets are in transit on the network at one time, which increases throughput by reducing or eliminating the time that either TCP endpoint spends waiting for new data after it has acknowledged what it has already received.

For more information about large TCP windows, see RFC 1323, "TCP Extensions for High Performance."

Selective Acknowledgments

In typical TCP-based communication, acknowledgments are cumulative. TCP only acknowledges segments received that are contiguous with previously acknowledged segments. Noncontiguous segments—segments received out of sequence—are not explicitly acknowledged. TCP requires that segments are received and acknowledged within a brief time period, or the missing segment, and all subsequent segments that follow it, must be retransmitted.

Selective acknowledgments are a recent TCP option that allow the receiver to selectively notify and request that a sender resend only data that is actually missing. This results in smaller amounts of data that require retransmission and better use of network bandwidth, provided that both TCP endpoints support this option.

For more information about selective acknowledgments, see RFC 2018, "TCP Selective Acknowledgment Options."

Better RTT Estimation

Round trip time (RTA) is used by TCP to estimate the amount of time that is needed for round trip communication between a sender and a receiver. Windows 2000 supports the use of the TCP Round Trip Time Measurement option to improve how RTT is estimated. By calculating more accurate RTT information more often, TCP uses better estimates for setting retransmission timers, which helps improve overall TCP speed and performance.

Improvements in estimating RTT help significantly when TCP connections include long-delay network links, such as long-distance or intercontinental WAN links, or wireless or satellite links.

For more information about TCP round trip time measurement, see RFC 1323.

ICMP Router Discovery

Internet Control Message Protocol Router Discovery messages are used to discover the default gateway on a network segment when a default gateway is either not manually configured or was not specified by DHCP. ICMP Router Discovery is implemented by use of two ICMP message types: the router solicitation and the router advertisement. A router solicitation packet is sent by a host when it needs to discover the routers on the network. Routers send router advertisement packets either on demand (i.e., in response to a router solicitation) or periodically (to notify hosts on the network that it is available). ICMP Router Discovery operates by default (as necessary) on TCP/IP for Windows 2000 hosts.

When a Windows 2000 Server computer (running the Routing and Remote Access service) is operating as a router, it can be configured to support the ICMP Router Discovery protocol as a router (e.g., it will send router advertisement messages when appropriate).

ICMP Router Discovery is described in RFC 1256, "ICMP Router Discovery Messages."

DNS Caching

The Windows 2000 DNS name resolver caches DNS name queries, and the cache can be viewed with the *ipconfig* utility. For specific information on the ipconfig command, refer to Appendix E.

Disabling NetBIOS over TCP/IP

With Windows 2000, you can disable NetBIOS over TCP/IP (NetBT) for each network connection. This feature is intended for computers that use only DNS name registration and resolution techniques and that communicate by using

the Client for Microsoft Networks and the File and Print Sharing for Microsoft Networks components with other computers where NetBT is disabled. Examples of disabling NetBT include computers in specialized or secured roles for your network, such as an edge proxy server or bastion host in a firewall environment, where NetBT support is not required or desired.

The following are considerations for disabling NetBT on computers running Windows 2000:

- The computer no longer listens for traffic to the NetBIOS datagram service at UDP port 138, the NetBIOS name service at UDP port 137, and the NetBIOS session service at TCP port 139.
- TCP/IP-based connections that use the Client for Microsoft Networks and the File and Print Sharing for Microsoft Networks components are possible only to other computers that have NetBT disabled. This affects the ability to see computers on the network and connect to file shares and network printers.
- NetBIOS name resolution techniques, such as Windows Internet Name Service (WINS), local subnet broadcasts, and the LAN Manager Hosts (Lmhosts) file, are no longer used. All name resolution occurs through DNS queries and the Hosts file.
- If the computer needs to participate in WINS as a client, it must have NetBT enabled on at least one network connection.
- If a Windows 2000 Server computer needs to run the WINS service, it must have NetBT enabled on at least one network connection.

A good example of disabling NetBT is for a server computer that has a connection to a private network and a connection to an external network, such as the Internet. In this case, NetBT is not required for the Internet connection. By disabling NetBT on only the Internet connection, the dual-homed computer continues to function as either a WINS server or client for the internal network, and WINS clients are still serviced for connections made by using other physical network adapters installed on the computer.

What Is A Dual-Homed System? The dual-homed system is a computer that includes at least two network interface cards. Windows 2000 supports this configuration, and you can enable or disable routing between the cards, depending on your requirements. Routing is disabled between the network interface cards in the dual-homed system so that the application-level software can control how traffic is handled between networks.

There is one other use for this type of configuration. Assume that the dual-homed host runs an HTTP Web service. If routing is disabled, then the host on either network can still access the Web services, but packets cannot be exchanged between the networks. For example, if several departments in an organization need to share the same Web server but you don't want to create a routable link between the departments, you could use this configuration.

You can disable NetBT on the **WINS** tab in the properties of the TCP/IP protocol. For information about configuring TCP/IP to use WINS, see Chapter 2, "Installing, Managing, and Diagnosing TCP/IP." A vendor-specific DHCP option has been defined by Microsoft that allows NetBT to be enabled or disabled by the DHCP server.

Additional Feature Information

For additional information about TCP/IP performance enhancements and features in Windows 2000, see the RFC documents previously listed. For information about obtaining RFC documents, see Appendix A, "TCP/IP Resources."

Security Features of TCP/IP for Windows 2000

TCP/IP for Windows 2000 incorporates security features that provide protection of the TCP/IP data as it is sent on the network and configuration of the types of local host traffic that are processed.

Internet Protocol Security

Internet Protocol security (IPSec) is a set of Internet standards employing cryptographic algorithms to provide the following:

- *Confidentiality*. IPSec traffic may be encrypted. Such traffic, when captured by a third party, is unintelligible without knowledge of the mutually determined *key* that each endpoint used to encrypt the traffic.
- *Authentication*. IPSec traffic may be digitally signed with a mutually determined authentication key so that either IPSec peer can verify that every authenticated packet was sent by the other IPSec peer. Packets can be authenticated without being encrypted.
- *Data integrity*. IPSec traffic contains a cryptographic checksum that incorporates the encryption key. The receiver can verify that the packet was not modified in transit.

For information about configuring IPSec policies for TCP/IP, see Chapter 2.

TCP/IP Filtering

With TCP/IP filtering, a feature known as TCP/IP Security in Windows NT version 4.0, you can specify exactly which types of incoming TCP/IP traffic are processed for each IP interface. This feature is designed to isolate the traffic that is processed by Internet or intranet servers in the absence of other TCP/

IP filtering provided by the Routing and Remote Access service or other TCP/IP applications or services. TCP/IP filtering is disabled by default. This service effectively allows a limited "firewall" capability to be placed *inside* the TCP/IP stack.

TCP/IP filtering is a set of filters for inbound local host TCP/IP traffic. Local host traffic is traffic that is processed by the host because the destination IP address of inbound TCP/IP traffic is addressed to an assigned interface address, an appropriate subnet broadcast address, or a multicast group address that this end-station has joined. TCP/IP filtering does not apply to routed traffic that is forwarded between interfaces (this only matters for systems with multiple interfaces that run the Routing and Remote Access service and have been configured to act as routers).

With TCP/IP filtering, the local host's inbound TCP/IP traffic can be selectively permitted or denied based on each packet's:

- Destination TCP port
- Destination UDP port
- IP protocol

For information about configuring TCP/IP filtering, see Chapter 2.

Core Protocols of TCP/IP

This section covers

- Address Resolution Protocol
- Internet Protocol
- Internet Control Message Protocol
- Internet Group Management Protocol
- User Datagram Protocol
- Transmission Control Protocol

Address Resolution Protocol

Address Resolution Protocol is a TCP/IP standard defined in RFC 826, "Address Resolution Protocol (ARP)." ARP resolves IP addresses used by TCP/IP-based software to media access control (MAC) layer addresses used by LAN hardware. ARP provides the following protocol services to hosts located on the same physical network:

- MAC addresses are obtained by using a subnetwork broadcast request in the form of the question, What is the MAC address for a device that is currently using the enclosed IP address?
- When an ARP request is answered, both the sender of the ARP reply and the original ARP requester record each other's IP address and

media access control address as an entry in a local table called the ARP cache for future reference. Often, systems that hear the original ARP request also record that system's MAC and IP addresses, so that if they need to communicate with that IP address in the future, they will not need to send an ARP request before doing so.

HARDWARE ADDRESSING

Hardware built for use on LANs must contain a unique address programmed into the device by the manufacturer. For Ethernet and Token Ring LAN hardware, this address is known as a MAC address.

Each MAC address identifies the device within its own local physical subnetwork. The MAC address takes the form of a 6-byte number programmed into the read-only memory (ROM) of each physical hardware device, such as an Ethernet or Token Ring network adapter. MAC addresses are typically displayed in hexadecimal notation (for example, 00-AA-00-3F-89-4A).

Registration of MAC addresses is overseen under the authority of the Institute of Electrical and Electronics Engineers (IEEE). Currently, the IEEE registers and assigns unique numbers for the first three bytes of the media access control address to individual manufacturers. Each manufacturer can then assign the last three bytes of the media access control address to individual network adapters.

HOW ARP RESOLVES MAC ADDRESSES FOR LOCAL TRAFFIC

Figure 1-4 shows how ARP resolves IP addresses to MAC addresses for hosts on the same local subnetwork.

In this example, two TCP/IP hosts, Hosts A and B, are both located on the same physical network. Host A is assigned the IP address of 10.0.0.99 and Host B is assigned the IP address of 10.0.0.100.

When Host A tries to communicate with Host B, the following steps resolve Host B's software-assigned address (10.0.0.100) to Host B's hardware-assigned MAC address:

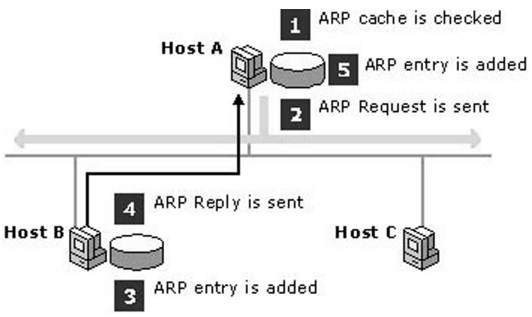

FIGURE 1–4 *How ARP resolves media access control addresses for local traffic*

1. Based on the contents of the routing table on Host A, IP determines that the forwarding IP address to be used to reach Host B is 10.0.0.100. Host A then checks its own local ARP cache for a MAC address matching Host B's IP address.
2. If Host A finds no mapping in the cache, it broadcasts an ARP request frame to all hosts on the local network with the question, What is the MAC address for 10.0.0.100? Both MAC and IP addresses for the source, Host A, are included in its ARP request.

   Each host on the local network receives the ARP request and checks for a match to its own IP address. If a host does not find a match, it discards the ARP request.
3. Host B determines that the IP address in the ARP request matches its own IP address and adds a MAC/IP address mapping for Host A to its local ARP cache.
4. Host B sends an ARP reply message containing its own MAC address directly back to Host A.
5. When Host A receives the ARP reply message from Host B, it updates its ARP cache with a MAC/IP address mapping for Host B.

Once the MAC address for Host B has been determined, Host A can send IP traffic to Host B by encapsulating the IP packets destined for Host B in MAC-layer frames destined for Host B's MAC address.

HOW ARP RESOLVES MAC ADDRESSES FOR REMOTE TRAFFIC

ARP is also used to forward IP datagrams to local routers for destinations that are not on the local subnetwork. In this situation, ARP resolves the MAC address of a router interface on the local network, usually the host's *default gateway*, which is either manually defined or learned by DHCP.

Figure 1-5 shows how ARP resolves IP addresses to MAC addresses for two hosts on different physical subnetworks connected by a common router.

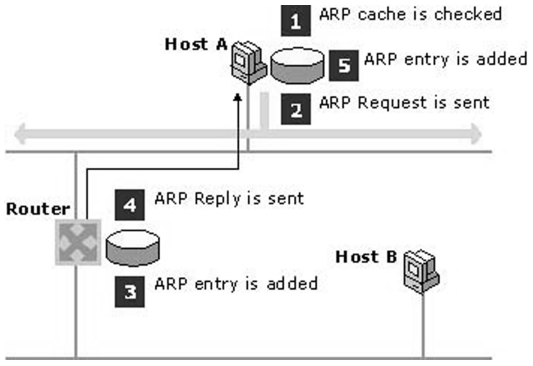

FIGURE 1–5 *How ARP resolves media access control addresses for remote traffic*

In this example, Host A is assigned an IP address of 10.0.0.99 and Host B uses an IP address of 192.168.0.99. Router interface 1 is on the same physical network as Host A and uses the IP address 10.0.0.1. Router interface 2 is on the same physical network as Host B and uses the IP address 192.168.0.1.

When Host A tries to communicate with Host B, the following steps resolve Router interface 1's software-assigned address (10.0.0.1) to its hardware-assigned MAC address:

1. Based on the contents of the routing table on Host A, IP determines that the forwarding IP address to be used to reach Host B is 10.0.0.1, the IP address of its default gateway. Host A then checks its own local ARP cache for a matching hardware address for 10.0.0.1.
2. If Host A finds no mapping in the cache, it broadcasts an ARP request frame to all hosts on the local network with the question, What is the MAC address for 10.0.0.1? Both MAC and IP addresses for the source, Host A, are included in the ARP request.

   Each host on the local network receives the ARP request and checks for a match to its own IP address. If a host does not find a match, it discards the ARP request.
3. The router determines that the IP address in the ARP request matches its own IP address and adds a MAC/IP address mapping for Host A to its local ARP cache.
4. The router then sends an ARP reply message containing its MAC address directly back to Host A.
5. When Host A receives the ARP reply message from the router, it updates its ARP cache with a MAC/IP address mapping for 10.0.0.1.

Once the MAC address for Router interface 1 has been determined, Host A can send IP traffic to Router interface 1 by encapsulating it in a frame destined for Router interface 1's MAC address. The router then forwards the traffic to Host B through the same ARP process as discussed in the previous section, "How ARP Resolves MAC Addresses for Local Traffic."

THE ARP CACHE

To minimize the number of broadcasts, ARP maintains a cache of IP address-to-MAC address mappings for future use. The ARP cache can contain both dynamic and static entries. Dynamic entries are added and removed automatically over time. Static entries remain in the cache until the computer is restarted.

Each dynamic ARP cache entry has a potential lifetime of 10 minutes. New entries added to the cache are time stamped. If an entry is not reused within two minutes of being added, it expires and is removed from the ARP cache. If an entry is used, it receives two more minutes of lifetime. If an entry

keeps getting used, it receives an additional two minutes of lifetime up to a maximum lifetime of 10 minutes.

Author's Note That a dynamic ARP cache entry has a potential lifetime of 10 minutes is OS-specific. The ARP specification does not mandate a cache lifetime. The fact that Windows 2000 does this is just silly.

You can view the ARP cache by using the **arp** command. To view the ARP cache on a computer running Windows 2000, type *arp -a* at a Windows 2000 command prompt. To view **arp** command-line options, type *arp /?* at a command prompt.

Note There is a separate ARP cache for each network adapter on a computer running Windows 2000.

Internet Protocol

IP is a TCP/IP standard defined in RFC 791, "Internet Protocol (IP)." IP is a connectionless datagram protocol primarily responsible for addressing and routing packets between hosts.

Connectionless means that packets can be sent into the network without requiring a per-destination setup procedure. Unreliable means that delivery is not guaranteed, yet IP always makes a best-effort attempt to deliver a packet. An IP packet might be lost, delivered out of sequence, duplicated, or delayed. IP does not attempt to recover from these types of errors. The acknowledgment of packets delivered and the recovery of lost packets is the responsibility of a higher-layer protocol, such as TCP.

An IP packet, also known as an *IP datagram*, consists of an IP header and an IP payload, as shown in Figure 1-6.

Table 1.5 lists the fields for addressing and routing contained in the IP header.

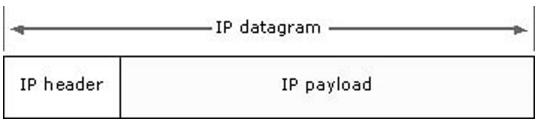

FIGURE 1–6 *An IP datagram*

TABLE 1.5 *IP Header Fields*

IP header field	Function
Source IP address	The IP address of the original source of the IP datagram.
Destination IP address	The IP address of the final destination of the IP datagram.
Time-to-Live (TTL)	Designates the number of network segments on which the datagram is allowed to travel before being discarded by a router. The TTL is set by the sending host and is used to prevent packets from endlessly circulating on an IP internetwork. When forwarding an IP packet, routers are required to decrease the TTL by at least 1.

Internet Control Message Protocol

Internet Control Message Protocol is a TCP/IP standard defined in RFC 792, "Internet Control Message Protocol (ICMP)." With ICMP, hosts and routers that use IP communication can report errors and exchange limited control and status information.

ICMP messages are usually sent automatically in one of the following situations:

- An IP datagram cannot reach its destination.
- An IP router (gateway) cannot forward datagrams at the current rate of transmission.
- An IP router redirects the sending host to use a better route to the destination.

ICMP messages are encapsulated within IP datagrams, as shown in Figure 1-7.

Different types of ICMP messages are identified in the ICMP header. Because ICMP messages are carried in IP datagrams, they are unreliable.

The most common ICMP messages are listed and described in Table 1.6.

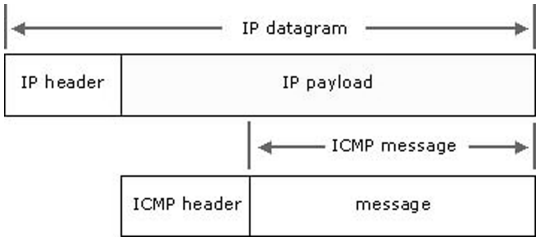

FIGURE 1–7 *ICMP encapsulation in an IP datagram*

TABLE 1.6 *ICMP Messages*

ICMP message	Description
Echo request	Determines whether an IP node (a host or a router) is available on the network.
Echo reply	Replies to an ICMP echo request.
Destination unreachable	Informs the host that a datagram cannot be delivered.
Redirect	Informs the host of a preferred route.
Time exceeded	Indicates that the TTL of an IP datagram has expired.

You can use the **ping** command to send ICMP echo request messages and record the receipt of ICMP echo reply messages. With these messages, you can detect network or host communication failures and troubleshoot common TCP/IP connectivity problems.

For more information about ICMP, see RFC 792. For more information about obtaining RFCs, see Appendix A.

Internet Group Management Protocol

The use of IP multicasting in TCP/IP networks is defined as a TCP/IP standard in RFC 1112, "Internet Group Management Protocol (IGMP)." In addition to defining address and host extensions for how IP hosts support multicasting, this RFC also defines the Internet Group Management Protocol version 1. RFC 2236, "Internet Group Management Protocol (IGMP), version 2" defines IGMP version 2. Both versions of IGMP provide a protocol to exchange and update information about host membership in specific multicast groups.

WHAT IS IP MULTICASTING?

Multicast IP traffic is sent to a single address but is processed by multiple hosts. IP multicasting is similar to a newsletter subscription. Just as only subscribers receive the newsletter when it is published, only host computers belonging to the multicast group receive and process IP traffic sent to the group's selected IP address. The set of hosts listening on a specific IP multicast address is called a *multicast group*.

Other important aspects of IP multicasting include the following:

- Group membership is dynamic, allowing hosts to join and leave the group at any time.
- The ability of hosts to join multicast groups is performed through the sending of IGMP messages.
- Groups are not limited by size and members can be spread out across multiple IP subnetworks (if a connected set of intervening routers support the propagation of IP multicast traffic).

- Any host can send IP traffic to any group's IP address without being a member of the corresponding group.

MULTICAST ADDRESSING

IP multicast addresses are reserved and assigned from within the Class D address range from 224.0.0.0 through 239.255.255.255. Table 1.7 is a partial list of well-known Class D addresses used by Windows 2000 components that are reserved for IP multicasting and registered with IANA.

TABLE 1.7 *Class D Addresses Used by Windows 2000*

IP multicast address	Description
224.0.0.0	Base address (reserved).
224.0.0.1	The All Hosts multicast group that contains all systems on the same network segment.
224.0.0.2	The All Routers multicast group that contains all routers on the same network segment.
224.0.0.5	The Open Shortest Path First (OSPF) AllSPFRouters address. Used to send OSPF routing information to all OSPF routers on a broadcast-capable subnetwork segment.
224.0.0.6	The OSPF AllDRouters address. Used to send OSPF routing information to OSPF designated routers on a broadcast-capable subnetwork segment.
224.0.0.9	The Routing Information Protocol (RIP) version 2 group address. Used to send RIP routing information to all RIP v2 routers on a network segment.
224.0.1.24	WINS server group address. Used to support autodiscovery and dynamic configuration of replication for WINS servers. For more information, see "WINS Replication" in Chapter 6, "WINS Overview."

A single IP address within the Class D reserved range identifies each multicast group. Each group's reserved IP address is shared by all host members of the group who listen and receive any IP messages sent to the group's IP address.

Author's Note Except for 224.0.1.24, none of these addresses are joined by hosts. And although, strictly speaking, it is illegal, the 224.0.0.x (where x is not 1) addresses are usually not joined by IGMP reports, even though RFC 1112 says they should be explicitly joined.

IP multicast addresses are mapped to a reserved set of media access control multicast addresses.

IP multicast is an open, IETF (Internet Engineering Task Force) standard for distributing data to multiple recipients. The multicast recipient group can change dynamically. A host may decide to join or leave a group at any time and a host may be a member of more than one multicast group. In addition,

any host can be a multicast source by simply sending packets addressed to a particular multicast group.

Routers in this scheme must be multicast enabled. When a multicast source transmits a multicast datagram, the local router forwards the packets to other routers with attached networks that include members of the multicast group.

IP multicast uses class D addressing, which is a special form of the IP address designed for multicasting. All hosts connected to the Internet have an IP address that is either part of the class A, class B, or class C scheme. A host can also have one or more class D multicast addresses, depending on the multicast groups it wants to belong to. A class D address is 32 bits long. The first 4 bits are used to identify it as a class D address. The remaining 28 bits identify multicast groups.

A class D address can be compared to the channel number of a TV station. When you tune in to a particular class D address, you receive packets that are being multicast by other systems that multicast on the address.

An industry consortium called the IPMI (IP Multicast Initiative) is dedicated to advancing the deployment of IP multicast and making information about it available. The IPMI Web site is at *www.ipmulticast.com*.

Note Web addresses can change, so you might be unable to connect to the Web site or sites mentioned here.

IGMP MESSAGES

IGMP allows IP routers to learn which of their directly attached subnetworks contain active group members at any given point in time. Membership in a multicast group is reported by at least one member host per subnetwork, in response to group membership solicitations (i.e., IGMP queries) that are periodically issued by multicast routers.

IGMP message types are described in Table 1.8.

TABLE 1.8 *IGMP Message Types*

IGMP message type	Description
Host membership report	When a host joins a multicast group, it sends an IGMP host membership report message, declaring its membership in a specific host group. IGMP host membership report messages are also sent in response to IGMP host membership query messages that are periodically sent by a router.
Host membership query	Used by a multicast router to periodically poll a network for group members.
Leave group	Sent by a host when it leaves a host group. If the host pays attention, it may decide to not send this message unless it can tell that it is the only group member present on its subnetwork.

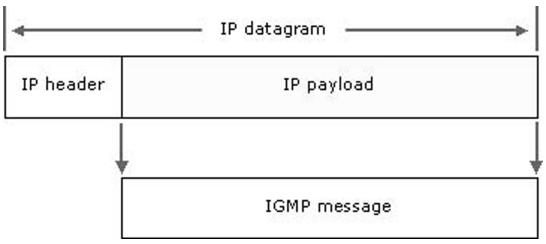

FIGURE 1–8 *GMP encapsulation in an IP datagram*

IGMP messages are encapsulated and sent within IP datagrams, as shown in Figure 1-8.

User Datagram Protocol

The User Datagram Protocol is a TCP/IP standard defined in RFC 768, "User Datagram Protocol (UDP)." UDP is used when applications require only the most basic transport-layer services. UDP functions mainly to allow multiple programs using different UDP ports to coexist on the same host. DNS, TFTP, Simple Network Management Protocol (SNMP), and Real-Time Protocol (RTP) are the principal uses of UDP.

UDP provides a connectionless datagram service that offers best-effort delivery, which means that UDP does not guarantee delivery or proper sequencing for any datagrams. A source host that needs reliable communication must use either TCP or an alternative transport protocol that provides its own sequencing and acknowledgment services.

UDP messages are encapsulated and sent within IP datagrams, as shown in Figure 1-9.

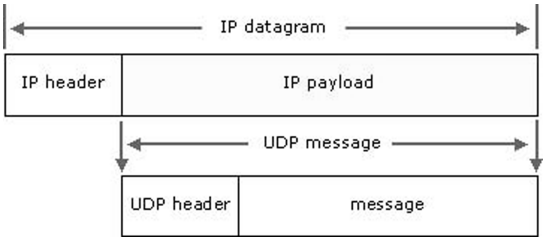

FIGURE 1–9 *UDP encapsulation in an IP datagram*

UDP PORTS

UDP ports provide a location for sending and receiving UDP messages. A UDP port functions as a single message queue for receiving all datagrams intended for the program specified by each protocol port number. This means UDP-based programs can receive more than one message at a time.

The server side of each program that uses UDP listens for messages arriving on its well-known port number. All UDP server port numbers less than 1,024 (and some higher numbers) are reserved and registered by IANA.

A well-known, or reserved, port number identifies every UDP-based program. Table 1.9 shows a partial list of well-known UDP server port numbers that are used by standard UDP-based programs.

TABLE 1.9 *UDP Server Port Numbers Used by Standard UDP-based Programs*

UDP port number	Description
53	DNS name queries
69	Trivial File Transfer Protocol
137	NetBIOS name service
138	NetBIOS datagram service
161	Simple Network Management Protocol
520	Routing Information Protocol (RIP)

UDP AND TCP

In general, differences in how UDP and TCP deliver data are similar to the differences between a telephone call and a postcard. TCP works like a telephone call by verifying that the destination is available and ready to communicate. UDP works like a postcard—messages are small and delivery is likely, but not always assured.

UDP is typically used when programs only need to transmit small amounts of data at a time, or by programs that have real-time requirements. In these situations, UDP's low overhead and multicasting potential (for example, one datagram, many recipients) may be a better match for certain applications than is TCP.

UDP contrasts directly with the services and features provided by TCP. Table 1.10 compares differences in how TCP/IP communication is handled depending on whether UDP or TCP is used for transporting data.

TABLE 1.10 *Differences in TCP/IP Communications Handling*

UDP	TCP
Connectionless service; no session is established between hosts.	Connection-oriented service; a session is established between hosts.
UDP does not guarantee or acknowledge delivery, or sequence data.	TCP guarantees delivery through the use of acknowledgments and sequenced delivery of data.
Programs that use UDP are responsible for providing any reliability needed to transport data.	Programs that use TCP are provided assurance of reliable data transport.
UDP is fast, has low overhead requirements, and can support point-to-point and point-to-multipoint communication.	TCP is slower, has higher overhead requirements, and only supports point-to-point communication.

Author's Note I am not propagating the falsehood that TCP is slower than UDP. This is simply not true. TCP can be just as fast as UDP. NFS was originally written with UDP, due to its perceived speed, but then NFS had to rather inelegantly reinvent a lot of the services that TCP does very efficiently. The author of NFS, years later, wished he had used TCP in the first place.

Both UDP and TCP use ports to identify communications for each TCP/IP program.

Transmission Control Protocol

Transmission Control Protocol is a TCP/IP standard defined in RFC 793, "Transmission Control Protocol (TCP)," that provides a reliable, connection-oriented packet delivery service. The Transmission Control Protocol:

- Guarantees delivery of IP datagrams.
- Performs segmentation and reassembly of large blocks of data sent by programs.
- Ensures proper sequencing and ordered delivery of segmented data.
- Performs checks on the integrity of transmitted data by using checksum calculations.
- Sends positive messages depending on whether data was received successfully. When using the optional selective acknowledgment extensions, negative acknowledgments for data not received may also be sent.
- Offers a preferred method of transport for programs that must use reliable session-based data transmission, such as client/server database and e-mail programs.

HOW TCP WORKS

TCP is based on point-to-point communication between two network hosts. TCP receives data from programs and processes this data as a stream of bytes. Bytes are grouped into segments that TCP then numbers and sequences for delivery.

Before two TCP hosts can exchange data, they must first establish a session with each other. A TCP session is initialized through a process known as a three-way handshake. This process synchronizes sequence numbers and provides control information that is needed to establish a virtual connection between both hosts.

Once the initial three-way handshake completes, segments are sent and acknowledged in a sequential manner between both the sending and receiving hosts. A similar handshake process is used by TCP before closing a connection to verify that both hosts are finished exchanging data.

TCP segments are encapsulated and sent within IP datagrams, as shown in Figure 1-10.

TCP PORTS

TCP ports use a specific program port for delivery of data sent by using Transmission Control Protocol. TCP ports are more complex and operate differently from UDP ports.

While a UDP port operates as a single message queue and the network endpoint for UDP-based communication, the final endpoint for all TCP communication is a unique connection. Each TCP connection is uniquely identified by dual endpoints.

Each single TCP server port is capable of offering shared access to multiple connections because all TCP connections are uniquely identified by two pairs of IP address and TCP ports (one address/port pairing for each connected host).

TCP programs use reserved or well-known port numbers, as shown in Figure 1-11.

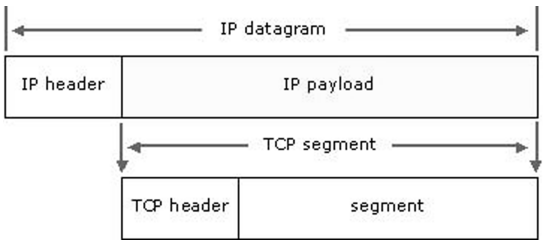

FIGURE 1–10 *TCP encapsulation in an IP datagram*

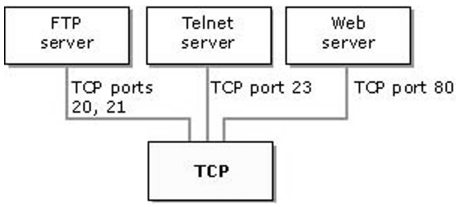

FIGURE 1–11 *TCP ports as used by programs*

The server side of each program that uses TCP ports listens for messages arriving on its well-known TCP port number. All TCP server port numbers less than 1,024 (and some higher numbers) are reserved and registered by IANA.

Table 1.11 is a partial list of some well-known TCP server ports used by standard TCP-based programs.

TABLE 1.11 *TCP Server Ports Used by Standard TCP-based Programs*

TCP port number	Description
20	FTP server (data channel)
21	FTP server (control channel)
23	Telnet server
25	Simple Mail Transport Protocol (SMTP)
53	Domain Name System zone transfers
80	Web server (HTTP)
139	NetBIOS session service

IP Addressing and Routing

This section covers:

- IP addressing
- Subnet masks
- IP routing
- The Windows 2000 IP routing table

Author's Note This section provides a general and brief overview of IP addressing and routing. For readers interested in learning more about this topic, especially beginners, I highly recommend IP Fundamentals: What Everyone Needs to Know About Addressing & Routing by Thomas A. Maufer.

Tom strikes a commendable balance between thoroughness and practicality. He makes sure—through liberal use of examples—that you understand everything necessary to complete your job. But he also goes to extraordinary lengths to bolster your grasp of the IP suite; his chapters conclude with fact-rich endnotes, plus references to journal articles and standards documents. In this way, IP Fundamentals helps you surpass mere technical competence to become an IP addressing and routing expert.

IP Addressing

Each TCP/IP host is identified by a logical IP address. This address is unique for each host that communicates by using TCP/IP. Each 32-bit IP address identifies a location of a host system on the network in the same way that a street address identifies a house on a city street.

Just as a street address has a standard two-part format (a street name and a house number), each IP address is separated internally into two parts—a network ID and a host ID:

- The network ID, also known as a network address, identifies a single network segment within a larger TCP/IP internetwork (a network of networks). All the systems that attach and share access to the same network have a common network ID within their full IP address. This ID is also used to uniquely identify each network within the larger internetwork.
- The host ID, also known as a host address, identifies a TCP/IP node (a workstation, server, router, or other TCP/IP device) within each network. The host ID for each device identifies a single system uniquely within its own network.

Here is an example of a 32-bit IP address:

```
10000011 01101011 00010000 11001000
```

To make IP addressing easier, IP addresses are expressed in dotted decimal notation. The 32-bit IP address is segmented into four 8-bit octets. The octets are converted to decimal (base-10 numbering system) and separated by periods. Therefore, the previous IP address example is 131.107.16.200 when converted to dotted decimal notation.

For more information about dotted decimal notation and the conversion of numbers from binary to decimal, see Appendix A.

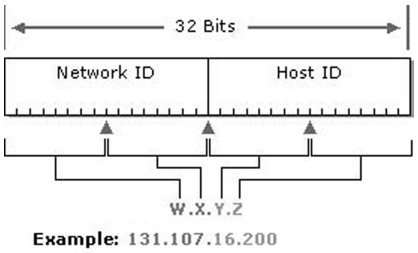

FIGURE 1–12 *The IP address: network ID and host ID*

Figure 1-12 shows a sample view of an IP address (131.107.16.200) as it is divided into network and host ID sections. The network ID portion (131.107) is indicated by the first two numbers of the IP address. The host ID portion (16.200) is indicated by the last two numbers of the IP address.

Note

Because IP addresses identify devices on a network, a unique IP address must be assigned to each device on the network.

In general, most computers have only a single network adapter installed and therefore require only a single IP address. If a computer has multiple network adapters installed, each adapter needs its own IP address.

IP ADDRESS CLASSES

The Internet community has defined five address classes. Class A, B, and C addresses are used for assignment to TCP/IP nodes.

The class of address defines which bits are used for the network and host ID parts of each address. The address class also defines how many networks and hosts per network can be supported.

Table 1.12 uses *w.x.y.z* to designate the four octet values in any given IP address. The table is used to show:

- How the value of the first octet (*w*) of any given IP address effectively indicates the class of address.
- How the octets in an address are divided into network ID and host ID.
- The number of possible networks and hosts per network available for each class.

TABLE 1.12 *Octet Values in an IP Address*

Class	Value of w	Network ID	Host ID	Number of networks	Maximum hosts per network
A	1-126	w	x.y.z	126	16,777,214
B	128-191	w.x	y.z	16,384	65,534
C	192-223	w.x.y	z	2,097,152	254
D	224-239	Reserved for multicast addressing	N/A	N/A	N/A
E	240-254	Reserved for experimental use	N/A	N/A	N/A

Subnet Masks

Network IDs and host IDs within an IP address are distinguished by using a subnet mask. Each subnet mask is a 32-bit number that uses consecutive bit groups of all ones (1) to identify the network ID and all zeros (0) to identify the host ID portions of an IP address.

For example, the subnet mask normally used with the IP address 131.107.16.200 is the following 32-bit binary number:

```
11111111 11111111 00000000 00000000
```

This subnet mask number is 16 one-bits followed by 16 zero-bits, indicating that the network ID and host ID sections of this IP address are both 16 bits in length. Normally, this subnet mask is displayed in dotted decimal notation as 255.255.0.0.

Table 1.13 displays subnet masks for the Internet address classes.

TABLE 1.13 *Subnet Masks for the Internet Address Classes*

Address class	Bits for subnet mask	Subnet mask
Class A	11111111 00000000 00000000 00000000	255.0.0.0
Class B	11111111 11111111 00000000 00000000	255.255.0.0
Class C	11111111 11111111 11111111 00000000	255.255.255.0

Typically, default subnet mask values (as shown in Table 1.13) are acceptable for most networks with no special requirements and where each IP network segment corresponds to a single physical network.

In some cases, you can use customized subnet masks to implement IP subnetting. With IP subnetting, you can subdivide the default host ID portion of an IP address to specify subnets, which are subdivisions of the original class-based network ID.

By customizing the subnet mask length, you can reduce the number of bits that are used for the actual host ID.

Important To prevent addressing and routing problems, you should make sure all TCP/IP computers on any network segment use the same subnet mask.

IP Routing

In general terms, routing is the process of forwarding packets between connected networks. For TCP/IP-based networks, routing is part of the Internet Protocol and is used in combination with other network protocol services to provide forwarding capabilities between hosts that are located on separate network segments within a larger TCP/IP-based network.

IP is the mailroom of the TCP/IP protocol, where IP data sorting and delivery take place. Each incoming or outgoing packet is called an *IP datagram*. An IP datagram contains two IP addresses: the source address of the sending host and the destination address of the receiving host. Unlike hardware addresses, the IP addresses within a datagram remain the same as it travels across a TCP/IP network.

Routing is the primary function of IP. IP datagrams are exchanged and processed on each host by using IP at the Internet layer.

Above the IP layer, transport services on the source host pass data in the form of TCP segments or UDP messages to the IP layer. The IP layer assembles IP datagrams with source and destination address information that is used to route the data through the network. The IP layer then passes datagrams down to the network interface layer. At this layer, data-link services convert IP datagrams into frames for transmission over network-specific media on a physical network. This process happens in reverse order on the destination host.

Each IP datagram contains a source and destination IP address. IP layer services on each host examine the destination address of each datagram, compare this address to a locally maintained routing table, and then decide what further forwarding action to take. IP routers are attached to two or more IP network segments and are enabled to forward packets between them. The following sections discuss IP routers and the use of routing tables in further detail.

IP ROUTERS

TCP/IP network segments are interconnected by IP routers. Routers are sometimes referred to as *gateways,* though that term is no longer in common use, except for the term *default gateway*, which you may have seen in your PC's IP configuration. Routers are devices that pass IP datagrams from one network or subnetwork to another. This process is known as IP routing and is shown in Figure 1-13.

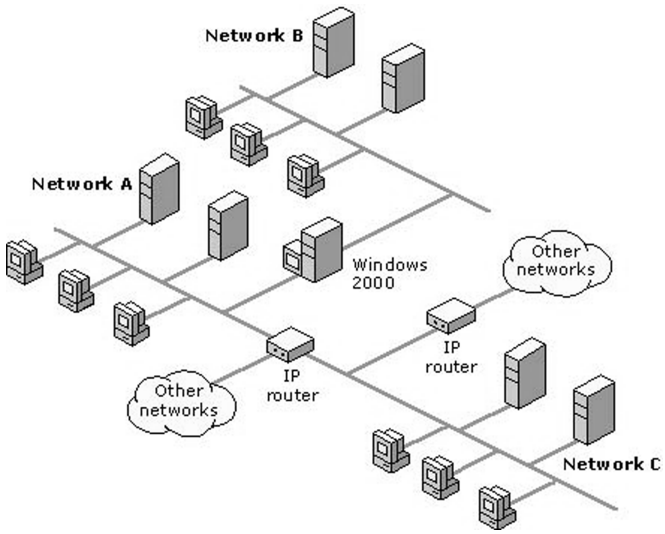

FIGURE 1–13 *IP routing*

IP routers provide the primary means of joining together two or more physically separated IP network segments. All IP routers share two essential characteristics:

- *IP routers are multihomed hosts*. A multihomed host is a network host that uses two or more network connection interfaces to connect to each physically separated network segment.
- *IP routers provide packet forwarding for other TCP/IP hosts*. IP routers are distinct from other hosts that use multihoming in one important way: an IP router must be able to forward IP-based communication between networks for other IP network hosts.

You can implement IP routers by using a variety of possible hardware and software products. Box-based routers—dedicated hardware devices that run specialized software—are common. In addition, you can use routing solutions that are based on software, such as the Routing and Remote Access service, which runs on a computer running Windows 2000 Server.

Regardless of the type of IP routers that you use, all IP routing relies on the use of a routing table to facilitate communication between network segments.

ROUTING TABLES

TCP/IP hosts use a routing table to maintain knowledge about other IP networks and IP hosts. Most hosts, such as the typical PC, have a very simple routing table, usually containing two entries. One of the entries represents the

PC's local subnetwork, while the other represents all other nonlocal networks. The latter entry is usually referred to as a *default route*. In certain circumstances, hosts may have more than just these two entries. Routers have more entries in their routing tables, including entries for the router's directly attached subnetworks, as well as all the destination subnetworks that the router has learned of. Each network or subnetwork and host is identified by an IP address and an associated subnet mask. In addition, routing tables are important because they provide needed information to each local host regarding how to communicate with remote networks and hosts.

For each computer on an IP network, you can maintain a routing table with an entry for every other computer or network that communicates with the local computer. In general, this is not practical, and a default gateway (IP router) is used instead.

When a computer prepares to send an IP datagram, it inserts its own source IP address and the destination IP address of the recipient into the IP header. The computer then examines the destination IP address, compares it to a locally maintained IP routing table, and takes appropriate action based on what it finds. The computer does one of three things:

- It passes the datagram up to a protocol layer above IP on the local host.
- It forwards the datagram through one of its attached network interfaces.
- It discards the datagram.

IP searches the routing table for the route that is the closest match to the destination IP address. The most specific to the least specific route is in the following order:

- A route that exactly matches the destination IP address (host route).
- A route that matches the network ID of the destination IP address (network route).
- The default route.

If a matching route is not found, IP discards the datagram.

The Windows 2000 IP Routing Table

Every computer that runs TCP/IP makes routing decisions. These decisions are controlled by the IP routing table. To display the IP routing table on a computer that is running Windows 2000, you can type *route print* at a command prompt.

Table 1.14 shows an example of an IP routing table. This example is for a computer running Windows 2000 with one network adapter and the following configuration:

- IP address: 10.0.0.169
- Subnet mask: 255.0.0.0
- Default gateway: 10.0.0.1

TABLE 1.14 *IP Routing Table Example*

Description	Network destination	Netmask	Gateway	Interface	Metric
Default route	0.0.0.0	0.0.0.0	10.0.0.1	10.0.0.169	1
Loopback network	127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
Local network	10.0.0.0	255.0.0.0	10.0.0.169	10.0.0.169	1
Local IP address	10.0.0.169	255.255.255.255	127.0.0.1	127.0.0.1	1
Multicast addresses	224.0.0.0	240.0.0.0	10.0.0.169	10.0.0.169	1
Limited broadcast address	255.255.255.255	255.255.255.255	10.0.0.169	10.0.0.169	1

Note The descriptions in the first column of Table 1.14 are not actually displayed in the output of the route print command.

The routing table is built automatically, based on the current TCP/IP configuration of your computer. Each route occupies a single line in the displayed table. The routing table is searched by your computer for an entry that is the most specific match to the destination IP address. The most specific matching routing table entry is the entry with the most leading bits in common with the destination IP address.

Your computer uses the default route if no other host or network route matches the destination address included in an IP datagram. The default route typically forwards an IP datagram (for which there is no matching or explicit local route) to a default gateway address for a router on the local subnet. In the previous example, the default route forwards the datagram to a router whose address is 10.0.0.1.

The default gateway contains information about the network IDs of the other IP subnets within the larger TCP/IP internetwork. It uses this information to forward the datagram to the next-hop gateway en route to the destination. The packet may need to cross many routers in order to reach its ultimate destination.

The following sections describe each of the columns displayed in the IP routing table: network destination, netmask, gateway, interface, and metric.

NETWORK DESTINATION

The network destination is used with the netmask to match the destination IP address. The network destination can range from 0.0.0.0 for the default route through 255.255.255.255 for the limited broadcast, which is a special broadcast address encompassing all hosts on the same network segment.

NETMASK

The netmask is the subnet mask that is applied to the destination IP address when matching to the value in the network destination. When netmask is written in binary, a "1" must match and a "0" need not match. For example, a 0.0.0.0 netmask is used for the default route, which means that none of the bits must match. For host routes, a route that represents just a single IP address, a netmask of 255.255.255.255 is used.

GATEWAY

The gateway address is the IP address that the local host uses to forward IP datagrams to other IP networks. This is either the IP address of a local network adapter or the IP address of an IP router (such as a default gateway router) on the local network segment.

INTERFACE

The interface is the IP address that is configured on the local computer for the local network adapter that is used when an IP datagram is forwarded on the network.

METRIC

A metric indicates the cost of using a route, which may equate to the number of hops to the IP destination. Whatever the "cost" represents, larger numbers generally mean that the associated network is further away. Anything on the local subnet is one hop, and each router crossed after that is an additional hop. If there are multiple routes to the same destination with different metrics, the route with the lowest metric is selected.

For information about adding and deleting routes in the IP routing table, see Chapter 2.

MULTIHOMED WINDOWS 2000 HOSTS

Table 1.15 shows the default routing table for a multihomed Windows 2000 host with this configuration:

- Network adapter 1
  - IP address: 10.0.0.169
  - Subnet mask: 255.0.0.0
  - Default gateway: 10.0.0.1
- Network adapter 2
  - IP address: 192.168.0.200
  - Subnet mask: 255.255.0.0
  - Default gateway: 192.168.0.1

TABLE 1.15 *Default Routing Table for a Multihomed Windows 2000 Host*

Network destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	10.0.0.1	10.0.0.169	1
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.200	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
10.0.0.0	255.0.0.0	10.0.0.169	10.0.0.169	1
10.0.0.169	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.0.0	192.168.0.200	192.168.0.200	1
192.168.0.200	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.0.255	255.255.255.255	192.168.0.200	192.168.0.200	1
224.0.0.0	240.0.0.0	10.0.0.169	10.0.0.169	1
224.0.0.0	240.0.0.0	192.168.0.200	192.168.0.200	1
255.255.255.255	255.255.255.255	10.0.0.169	10.0.0.169	1
255.255.255.255	255.255.255.255	192.168.0.200	192.168.0.200	1

For information about enabling IP forwarding on a multihomed computer running Windows 2000, see Chapter 2.

Note When you configure a default gateway to each network adapter, you create a 0.0.0.0 route for each network adapter. However, only one default route is actually used. In the previous example, the 10.0.0.169 IP address is the first network adapter in the TCP/IP bindings, and therefore the default route for Network adapter 1 is used. Because only one default gateway is used, you only need to configure one network adapter with a default gateway. This reduces confusion and assures the results you intended.

If the IP router is a Windows 2000 router and does not have an interface on a given network, it needs a route to get there. You can add static routes or use routing protocols that are provided by the Routing and Remote Access service.

Name Resolution

As we have seen, IP is designed to work with the 32-bit IP addresses of the source and the destination hosts. Computers are used by people who are not very good at using and remembering the IP addresses of the computers with which they want to communicate. People are much better at using and remembering names rather than IP addresses.

If a name is used as an alias for an IP address, you need to ensure that the name is unique and that it resolves to the correct IP address.

In Windows 2000, there are two types of names to resolve:

- *Host names*. Programs that use the Windows Sockets programming interface, such as Web browsers, use host names. For information, see Appendix D, "DNS Resources and Troubleshooting."
- *NetBIOS names*. NetBIOS names are used by network programs or services that use the NetBIOS programming interface, such as Client for Microsoft Networks and File and Printer Sharing for Microsoft Networks. For more information, see Appendix C, "WINS Resources and Troubleshooting."

TCP/IP Configuration Items

For Windows 2000 TCP/IP to function properly, you need to configure the following:

- IP address
- Subnet mask
- Default gateway
- DNS server
- WINS server

IP Address

Each interface on each TCP/IP node (host or router) must be configured with a unique IP address that is correct for the attached network segment. The IP address is a required configuration item.

For more information, see "IP Addressing" in this chapter.

Subnet Mask

Each interface on each TCP/IP node (host or router) must be configured with a subnet mask that, when combined with the IP address, yields the network ID. All IP interfaces on the same network segment must use the same network ID. Therefore, all IP interfaces on the same network segment must use the same subnet mask. The subnet mask is a required configuration item.

For more information, see "Subnet Masks" in this chapter.

Default Gateway

To communicate with TCP/IP nodes on other network segments, you must configure at least one interface with the IP address of a default gateway (a local router that forwards remote TCP/IP traffic to its destination).

You do not need to configure a default gateway for a network that consists of a single network segment.

For more information, see "IP Routing" in this chapter.

DNS Server

A DNS server can resolve domain names to IP addresses. When a TCP/IP host is configured with the IP address of a DNS server, the TCP/IP host sends DNS name queries to the DNS server for resolution. Computers running Active Directory-based Windows 2000 require that a DNS server be present.

You do not need to configure a DNS server for a network that consists of a single network segment.

A computer running Windows 2000 Server can be a DNS server. This is covered later in the book.

WINS Server

A WINS server can store and resolve NetBIOS names to IP addresses. When a TCP/IP host is configured with the IP address of a WINS server, the TCP/IP host registers its own NetBIOS names with the WINS server and sends NetBIOS name queries to the WINS server for resolution. A WINS server is highly recommended when there is more than one network segment in your network and when you have computers that are not based on Active Directory (for example, computers running Windows NT 4.0, Windows 95, and Windows 98).

You do not need to configure a WINS server for a network that consists of a single network segment.

A computer running Windows 2000 Server can be a WINS server. This is covered later in the book.

TCP/IP Configuration Methods

You can configure TCP/IP in Windows 2000 by using the following methods: automatic configuration, dynamic configuration, and manual configuration.

If you are using either a dynamic or manual configuration, find out more about developing an effective TCP/IP network numbering plan in "Numbering Your Network" in this chapter.

Automatic Configuration

By using the new Automatic Private IP Addressing feature (also known as DHCP Autoconfiguration), Windows 2000 provides default automatic configuration of the IP address within the range reserved for APIPA that runs from 169.254.0.1 through 169.254.255.254 and uses an associated subnet mask of 255.255.0.0. There is no automatic configuration of a default gateway, DNS server, or WINS server. APIPA is designed for networks that consist of a single network segment that are not connected to the Internet. Therefore, you do not need to configure the default gateway, DNS server, and WINS server.

For information about configuring TCP/IP for automatic addressing, see Chapter 2.

Dynamic Configuration

By using DHCP, TCP/IP configuration is done dynamically and automatically when the computer is started. Dynamic configuration requires the configuration of a DHCP server. By default, computers running Windows 2000 are DHCP clients. By properly configuring the DHCP server, TCP/IP hosts can obtain IP address, subnet mask, default gateway, DNS server, NetBIOS node type, and WINS server configuration information. Dynamic configuration (using DHCP) is recommended for medium-to-large TCP/IP networks.

For information about configuring TCP/IP for dynamic addressing, see Chapter 2.

Manual Configuration

By manually configuring the properties of the TCP/IP protocol through the properties of a network connection (in Network and Dial-up Connections), you can assign an IP address, subnet mask, default gateway, DNS server, and WINS server. Manual configuration is required in a network with multiple network segments when no DHCP server is present.

For information about configuring TCP/IP for static addressing, see Chapter 2.

Installing TCP/IP

Before you install Microsoft TCP/IP on a computer running Windows 2000, you need to know:

- Whether your network supports a *dynamic* TCP/IP configuration, which uses the Dynamic Host Configuration Protocol (DHCP). A

dynamic configuration is the simplest to complete and can be used if another computer on your network is installed as a DHCP server. The DHCP server provides IP address, subnet mask, default gateway (IP router), DNS domain name, DNS server, and WINS server configuration information.

- Whether your network requires a *manual* TCP/IP configuration. Some existing networks either do not use DHCP or require that TCP/IP configuration is performed manually for each computer.

For more information about configuration methods, see "TCP/IP Configuration Methods" in this chapter.

If you do not use DHCP to configure TCP/IP dynamically, obtain the following information from your network administrator to configure TCP/IP manually:

- The IP address and subnet mask for each network adapter that is installed on the computer.
- The IP address for the default local gateway (IP router).
- The name of your DNS domain and the IP addresses of the DNS servers on the network.

  For Active Directory deployment, the primary DNS suffix of a computer, also known as the computer's primary domain name, is by default the name of the Active Directory domain that the computer has joined. If you want your computer's primary DNS suffix to be different from the name of the Active Directory domain that the computer has joined, you must consult "To Change the DNS Suffix of Your Computer" in this chapter.
- The IP addresses for the WINS servers, if WINS services are available on your network.

For more information about configuration items, see "TCP/IP Configuration Items" in this chapter. For instructions about installing and configuring TCP/IP, see Chapter 2.

To Change the DNS Suffix of Your Computer

To change the DNS suffix of your computer, you must be logged on as an administrator to the local computer to change the DNS suffix:

1. Open System in Control Panel.

   To open a Control Panel item, click **Start**, point to **Settings**, click **Control Panel**, and then double-click the appropriate icon.
2. On the **Network Identification** tab, click **Properties**, and then click **More**.
3. Under **Primary DNS suffix of this computer**, type the DNS suffix you want to use.

The default setting for the local primary DNS suffix is the same as the Active Directory domain name. Changing the DNS suffix will not affect your domain membership, but it could prevent other users from finding your computer on the network. If you make the primary DNS suffix of the computer different from the Active Directory domain name, the domain administrator must enable registration of the new full computer name in the Active Directory domain.

A DNS suffix may contain up to 155 characters, including hyphens and periods.

4. If you want your computer to automatically update the DNS suffix if you become a member of a different Windows 2000 domain, select the **Change primary DNS suffix when domain membership changes** check box.

If you switch to a new Windows 2000 secure domain and select the **Change primary DNS suffix when domain membership changes** check box, the DNS suffix is updated to show a new DNS suffix that matches the new Windows 2000 secure domain you are joining. For example, if you create a DNS suffix called *MyMachine* for membership in your current domain and join a new Windows 2000 secure domain called *example.microsoft.com*, then the new DNS suffix (*example.microsoft.com*) is displayed under **Primary DNS suffix of this computer**. This new DNS suffix replaces the previous name *(MyMachine)* you created for membership under your old domain.

If this computer belongs to a group with a group policy enabled on **Primary DNS suffix of this computer**, the string specified in the group policy is used as the primary DNS suffix. The local setting is used only if a group policy is disabled or unspecified.

Numbering Your Network

For private TCP/IP networks that are indirectly connected to the Internet by using a network address translator (NAT) or an application layer gateway such as a proxy server, IANA recommends that you use the private IP addresses shown in Table 1.16.

TABLE 1.16 *Recommended Private IP Addresses*

Private network ID	Subnet mask	Range of IP addresses
10.0.0.0	255.0.0.0	10.0.0.1 - 10.255.255.254
172.16.0.0	255.240.0.0	172.16.0.1 - 172.31.255.254
192.168.0.0	255.255.0.0	192.168.0.1 - 192.168.255.254

Numbers in these ranges are reserved by IANA for private use on TCP/IP networks and are not used on the Internet.

For security reasons, it is unlikely that you would need to connect more than a few TCP/IP systems within your network directly to the Internet. For any host systems on your network that connect to the Internet, you must obtain the use of registered IP addresses from your Internet service provider (ISP).

Note

If your ISP cannot provide sufficient public addresses to accommodate your needs and you are located in the United States, you can apply directly to the American Registry for Internet Numbers (ARIN) to obtain an IANA-assigned IP network address. For more information about registering a public address, see the ARIN Web site (www.arin.net/intro.html).

Web addresses can change, so you might be unable to connect to the Web site or sites mentioned here.

For more information about private IP address numbering for private networks, see RFC 1918, "Address Allocation for Private Internets."

Default Gateways

Default gateways serve an important role in TCP/IP networking. They provide a default route for TCP/IP hosts to use when communicating with other hosts on remote networks.

Figure 1-14 shows the roles played by the default gateways (IP routers) for two networks: Network 1 and Network 2.

In order for Host A on Network 1 to communicate with Host B on Network 2, Host A first checks its routing table to see if a specific route to Host B exists. If there is no specific route to Host B, Host A forwards its TCP/IP traffic for Host B to its own default gateway, IP Router 1.

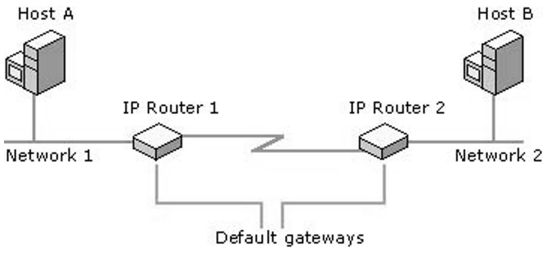

FIGURE 1–14 *Role of default gateways*

The same principle applies if Host B is sending to Host A. Without a specific route to Host A, Host B forwards any TCP/IP traffic destined for Host A to its own default gateway, IP Router 2.

Why Gateways Work

Default gateways are important to make IP routing work efficiently. In most cases, the router that acts as the default gateway for TCP/IP hosts—either a dedicated router or a computer that connects two or more network segments—maintains knowledge of other networks in the larger network and how to reach them.

TCP/IP hosts rely on default gateways for most of their communication needs with hosts on remote network segments. In this way, individual hosts are freed of the burden of having to maintain extensive and continuously updated knowledge about individual remote IP network segments. Only the router that acts as the default gateway needs to maintain this level of routing knowledge to reach other remote network segments in the larger internetwork.

If the default gateway fails, communication beyond the local network segment may be impaired. To prevent this, you can use the **Advanced TCP/IP Settings** dialog box (in Network and Dial-up Connections) for each connection to specify multiple default gateways. You can also use the **route** command to manually add routes to the routing table for heavily used hosts or networks.

Using Multiple Gateways

When configuring TCP/IP for Windows 2000, you can add more default gateways for each network adapter. When a computer is configured for multiple default gateways, all remote TCP/IP traffic that does not match an entry in the routing table is passed to the first default gateway defined for the first network adapter.

Because only one default gateway (the one defined for the first network adapter) is used, you should configure only one network adapter to have default gateways. This practice reduces confusion and assures the results you intended. For example, if you have two network adapters and configure a default gateway for each, the default gateway of the first network adapter is used. The default gateway for the second is used if the first is unavailable.

Author's Note The "multiple default gateways" feature is proprietary to Microsoft dead gateway detection and only works for TCP-based applications.

Advanced Configuration

Beyond the configuration of an IP address, subnet mask, default gateway, DNS server, and WINS server, you can configure Windows 2000 TCP/IP for:

- Advanced IP settings
- Advanced DNS settings
- Advanced WINS settings
- Advanced options

Advanced IP Settings

Advanced IP settings include:

- *Multiple IP addresses*. You can use multiple IP addresses per network adapter to allow for implementation of multiple IP numbering schemes, such as public addresses used for the Internet and private addresses, and for multiple logical IP networks on the same physical network segment.
- *Multiple default gateways*. You can use TCP/IP for Windows 2000 to detect downed routers by using a feature called *dead gateway detection*. If multiple default gateways are configured, a failing TCP connection updates the IP routing table with the next default gateway in the list.

For more information about configuring advanced IP settings, see Chapter 2.

Advanced DNS Settings

Advanced DNS settings include:

- *Multiple DNS servers*. If multiple DNS servers are configured and TCP/IP fails to receive any response from the current DNS server, TCP/IP switches to the next DNS server.
- *Unqualified name resolution*. For unqualified names, you can configure TCP/IP to either
  - Append the primary and connection-specific DNS suffixes to the unqualified name for DNS queries. For example, when looking up the machine named elvis, Windows 2000 completes the rest of the domain name by adding graceland.com for a fully-qualified name of elvis.graceland.com. Any unqualified name would have graceland.com appended to it.

- Append a series of configured DNS suffixes to the unqualified name for DNS queries. If the DNS resolver is presented with an unqualified domain name, it will try to add each configured domain extension and see if there is a match. For instance, if the system is looking up the domain name *corp-acctg,* the resolver will try appending *graceland.com* and look up *corp-acctg.graceland.com*. If that name does not exist, the system will then try appending *memphis.com* to see if *corp-acctg.memphis.com* exists, and if not, then it would try the next extension. If there were no matches, then the system would return an error indicating that the name does not exist. In that case, the user could try to be more specific and type the entire, fully qualified domain name (if they know it).

- *Connection-specific DNS suffixes*. You can configure each connection in Network and Dial-up Connections to have its own DNS suffix in addition to the primary DNS suffix that is configured for the computer on the **Network Identification** tab (available in System in Control Panel).

  Again, an example would be very helpful here. For instance, you could show how a computer that has a dial-up connection at home might also have an Ethernet connection at work. The DNS configuration for the corporate network would clearly need to be different than that of the dialup ISP.

- *DNS dynamic update behavior*. If you have DNS servers that support DNS dynamic update (such as the Windows 2000 DNS service), you can enable the DNS dynamic update of the domain name and IP addresses for the computer. If you configure a connection-specific DNS suffix, you can also enable the DNS dynamic update of the domain name and IP addresses for the connection.

For more information about configuring advanced DNS settings, see Chapter 2.

Advanced WINS Settings

Advanced WINS settings include:

- *Multiple WINS servers*. If multiple WINS servers are configured and TCP/IP fails to receive any response from the current WINS server, TCP/IP switches to the next WINS server.
- *Enabling and disabling the use of the Lmhosts file*. If the Lmhosts file is enabled, TCP/IP parses the Lmhosts file found in the *systemroot\System32\Drivers\Etc* folder during NetBIOS name resolution. By default, the use of the Lmhosts file is enabled. For more information, see Appendix A.

- *Enabling and disabling the use of NetBIOS over TCP/IP*. You can enable or disable the use of NetBIOS over TCP/IP. If the use of NetBIOS over TCP/IP is disabled, NetBIOS programs cannot run over TCP/IP. Therefore, you may not be able to connect to computers that are running an operating system other than Windows 2000. By default, the use of NetBIOS over TCP/IP is specified by the settings obtained from a DHCP server.

For more information about configuring advanced WINS settings, see Chapter 2.

Note You can no longer configure the NetBIOS scope ID on the WINS tab.To configure the NetBIOS scope ID, set the following registry value to the name of the scope ID that you want to use:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT\Parameters\ScopeID

Advanced Options

Advanced options include:

- *Enabling Internet Protocol security and specifying an IPSec security policy*. If IPSec is configured, secure end-to-end communication of IP-based traffic on a private network or the Internet is used. By default, IPSec is disabled. For more information, see Chapter 2.
- *Enabling TCP/IP filtering and specifying TCP/IP traffic*. If TCP/IP filtering is enabled, you can specify what types of TCP/IP traffic are processed. For more information, see Chapter 2.