NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

Transmission Control Protocol / Internet Protocol (TCP/IP)

Graduation Project
COM 400

Student: Banu İKİZLER

Supervisor:Asst.Prof.Dr.Firudin MURADOV

Nicosia - 2003

ACKNOWLEGMENT

"First, I would like to thank my supervisor Asst.Prof.Dr.Firudin MURADOV for his invaluable advice and belief in my work and myself over the course of this Graduation Project..

Second, I would like to express my gratitude to Near East University for the scholarship that made the work possible.

Third, I thank my family for their constant encouragement and support during the preparation of this project.

Finally, I would also like to thank all my friends for their advice and support."

ABSTRACT

Transmission Control Protocol/Internet Protocol for Microsoft Windos 2000 offers a standard, routable enterprise networking protocol that is the most complete and accepted protocol available. Most network operating systems in use today offer TCP/IP support, and large networks rely on TCP/IP for much of their network traffic.

TCP/IP provides a technology for connecting dissimilar systems. It is also provides a robust, scaleable, cross platform client/server framework and a foundation for gaining access to global Interner Services, such as the World Wide Web and e-mail.

The various protocols in the TCP/IP stack function together to make network communication happen. This process involves multiple activities, including resolving user-friendly computer names to Internet Protocol (IP) Adresses; determining the location of the destination computer; and packaging, adressing and routing the data so that it reaches the destination successfully. Internet protocol (IP) helps to identify the location of the destination computer in a network communication. The primary function of IP is to address information to data packets and route them accross the network to the correct destination.

TCP/IP transmits data on the network by dividing it into smaller portions called packets. The division of the data into packets is necessary because a large unit of data takes a long time to move on the network. As a packet of data moves from one layer to another in the TCP/IP stack. Each protocol adds it own header information. The packets of data being transmitted from one computer to another travel through the layers of the TCP/IP protocol stack

LIST OF ABBREVIATIONS

ARPANET Advanced Research Project Agency
ATM Asynhronous Transfer Mode
API Application Programming Interface
ARP Address Resolution Protocol
ACK Acknowledgment
CIDR Classless Inter-Domain Routing
DARPA Department of Defense Advanced Research Projects Agency
DNS Domain Name System
FTP File Transfer Protocol
FQDN Fully Qualified Domain Name
HTTP Hyper Text Transfer Protocol
ICMP Internet Control Message Protocol
IGMP Internet Group Management Protocol
IANA Interned Assigned Numbers Authority
InterNIC Internet Network Information Center
LAN Local Area Network
NETBıos Network Basic Input/Output System
NET Network Address Traslator
MTU Maximum Transmission Unit
MAC Media ACCESS control
OSI Open System Interconnection
OSPF Open Shortest Path First
RFC Request For Comments
RIP Routing Information Protocol
SNMP Simple Network Management Protocol
SMTP Simple Mail Transfer Protocol
TCP/IP Transmission Control protocol/Internet protocol
TFTP Trivial File Transfer Protocol
UDP User Datagram Protocol
WAN Wide Area Network
WINS Windows Internet Name Service

INTRODUCTION

Transmission Control Protocol/Internet Protocol is an industry standard protocol stack that is used for communication between Windows 2000 based computers. TCP/IP is designed for a communication across large scale networks. In Chapter one TCP/IP protocol suite enables enterprise networking and connectivity on windows 2000 based computers. A suite is created by a vendor or organization to customize a protocol stack for its requirements. Therefore, a protocol suite is a set of protocols designed and built as complementary parts of a complete, smoothly functioning set.

Chapter two describes the tasks involved in using TCP/IP in the communication process are distributed between protocols that are organized into four distinct layers of the TCP/IP stack. Each protocol in the TCP/IP stack has a distinct role in the communication process. During the communication process, many applications may be in communication at the same time. TCP/IP has the ability to differentiate one application from another. TCP/IP identifies an application on one computer and then moves the data from that application to an application on another computer.

Chapter three represents the primary function of Internet Protocol (IP) is to add address information to data packets and to route them accross the network to the correct destination. To understand how IP accomplishes this, it is necessary to be familiar with the concepts that determine the intermediate and final destination addresses of the data packet IP adresses are organized into classes. This is called classfull IP addressing. You must assign IP addresses to all computers that are connected on the network. Network segments connected by a router are called subnets. Breaking down the network into subnets for the purpose of assigning IP addresses is called subnetting.

Chapter four describes names. TCP/IP identifies source and destination computers by their IP addresses. However, users are much better at remembering and using words (user-friendly names) than numbers (IP adresses). There are different types of userfrinedly names by which a computer can be addressed. The windows 2000 operating system has different storage locations where it keeps a record of users-friendly names mapped to their corresponding IP addresses. This mapping of the IP address of a computer can be stored in either a static or dynamic file, based on the type of name used.

Chapter five describes the meaning of routers. Large TCP/IP networks, referred to as interworks, are broken up into smaller segments to reduce the amount of communication within the segment. A internetwork is a network consisting of multiple segments that are connected by routers. Routers are basically computers with two network adapters that provide the primary means of joiningtwo or more physically seperated segments. Routers pass IP packets from one network segment to another This process of forwarding IP packets as known as routing.

Chapter six is devoted to the internet Layer of the TCP/IP suite, Address Resolution Protocol performs address resolution for out going packets. Address resolution is the process by which IP addresses are mapped to MAC addresses. The network adapters use the MAC address to determine if a packet is meant for that computer.

CHAPTER ONE: THE TCP/IP PROTOCOL SUITE

TCP/IP is an industry-standard suite of protocols designed for large internet works spanning wide area network (WAN) links. TCP/IP was developed in 1969 by the U.S. Department of Defense Advanced Research Projects Agency (DARPA), as a result of the resource-sharing experiment called Advanced Research Projects Agency Network (ARPANET). The purpose of TCP/IP was to provide high-speed communication network links. Since 1969, ARPANET has grown into a worldwide community of networks known as the Internet.

1.1. TCP/IP for Windows 2000

TCP/IP for Windows 2000 enables enterprise networking and connectivity on Windows 2000–based computers. Adding TCP/IP to a Windows 2000 configuration offers the following advantages:

- A standard, routable enterprise networking protocol that is one of the most complete and accepted protocol available. All modern network operating systems offer TCP/IP support, and most large networks rely on TCP/IP for much of their network traffic.
- A technology for connecting dissimilar systems. Many standard connectivity utilities are available to access and transfer data between dissimilar systems, including File Transfer Protocol (FTP) and Telnet, a terminal emulation protocol. Several of these standard utilities are included with Windows 2000 Server.
- A robust, scaleable, cross-platform client-server framework. TCP/IP for Windows 2000 offers the Windows® Sockets interface, which is ideal for developing client-server applications that can run on Windows Sockets–compliant stacks from other vendors.
- A method of gaining access to the Internet. The Internet consists of thousands of networks worldwide connecting research facilities, universities, libraries, and private companies.

Note The word internet (lowercase i) refers to multiple TCP/IP networks connected with routers. References to the Internet (uppercase I) refer to the worldwide public Internet. References to an intranet refer to a private internetwork.

1.2. TCP/IP Standards

The standards for TCP/IP are published in a series of documents called Request for Comments (RFCs). RFCs describe the internal workings of the Internet. Some RFCs describe network services or protocols and their implementations, whereas others summarize policies. TCP/IP standards are always published as RFCs, although not all RFCs specify standards.

TCP/IP standards are not developed by a committee, but rather by consensus. Anyone can submit a document for publication as an RFC. Documents are reviewed by a technical expert, a task force, or the RFC editor, and then assigned a status. The status specifies whether a document is being considered as a standard.

There are five status assignments of RFCs as described in Table 1.

Table 1 Status assignments of RFCs

Status	Description
Required	Must be implemented on all TCP/IP-based hosts and gateways.
Recommended	Encouraged that all TCP/IP-based hosts and gateways implement the RFC specifications. Recommended RFCs are usually implemented.
Elective	Implementation is optional. Its application has been agreed to, but is not a requirement.
Limited Use	Not intended for general use.
Not recommended	Not recommended for implementation.

If a document is being considered as a standard, it goes through stages of development, testing, and acceptance known as the Internet Standards Process. These stages are formally labeled maturity levels. Table 2 lists the three maturity levels for Internet Standards.

Table 2 Maturity levels for Internet Standards

Maturity Level	**Description**
Proposed Standard	A Proposed Standard specification is generally stable, has resolved known design choices, is believed to be well understood, has received significant community review, and appears to enjoy enough community interest to be considered valuable.
Draft Standard	A Draft Standard must be well understood and known to be quite stable, both in its semantics and as a basis for developing an implementation.
Internet Standard	The Internet Standard specification (which may simply be referred to as a Standard) is characterized by a high degree of technical maturity and by a generally held belief that the specified protocol or service provides significant benefit to the Internet community.

When a document is published, it is assigned an RFC number. The original RFC is never updated. If changes are required, a new RFC is published with a new number. Therefore, it is important to verify that you have the most recent RFC on a particular topic.

RFCs can be obtained in several ways. The simplest way to obtain any RFC or a full and up-to-date indexed listing of all RFCs published is to access http://www.rfc-editor.org/rfc.html on the World Wide Web. RFCs can also be obtained by means of FTP from nis.nsf.net, nisc.jvnc.net, venera.isi.edu, wuarchive.wustl.edu, src.doc.ic.ac.uk, ftp.concert.net, internic.net, or nic.ddn.mil.

CHAPTER TWO: TCP/IP PROTOCOL ARCHİTECTURE

TCP/IP protocols map to a four-layer conceptual model known as the DARPA model, named after the U.S. government agency that initially developed TCP/IP. The four layers of the DARPA model are: Application, Transport, Internet, and Network Interface. Each layer in the DARPA model corresponds to one or more layers of the seven-layer Open Systems Interconnection (OSI) model.

Figure 1 shows the TCP/IP protocol architecture.

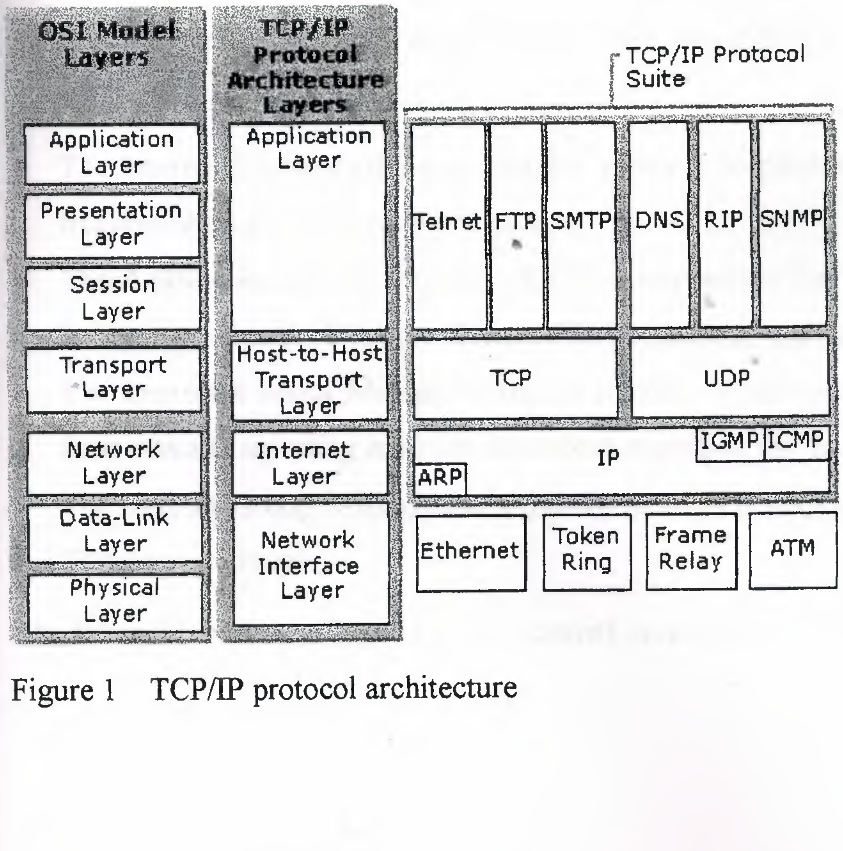


Figure 1 TCP/IP protocol architecture

2.1. Network Interface Layer

The Network Interface Layer (also called the Network Access Layer) is responsible for placing TCP/IP packets on the network medium and receiving TCP/IP packets off the network medium. TCP/IP was designed to be independent of the network access method, frame format, and medium. In this way, TCP/IP can be used to connect differing network types. This includes LAN technologies such as Ethernet or Token Ring and WAN technologies such as X.25 or Frame Relay. Independence from any specific network technology gives TCP/IP the ability to be adapted to new technologies such as Asynchronous Transfer Mode (ATM).

The Network Interface Layer encompasses the Data Link and Physical layers of the OSI Model. Note that the Internet Layer does not take advantage of sequencing and acknowledgment services that may be present in the Data Link Layer. An unreliable Network Interface Layer is assumed, and reliable communications through session establishment and the sequencing and acknowledgment of packets is the responsibility of the Transport Layer.

2.1.1 Internet Layer

The Internet Layer is responsible for addressing, packaging, and routing functions. The core protocols of the Internet Layer are IP, ARP, ICMP, and IGMP.

- The Internet Protocol (IP) is a routable protocol responsible for IP addressing and the fragmentation and reassembly of packets.
- The Address Resolution Protocol (ARP) is responsible for the resolution of the Internet Layer address to the Network Interface Layer address, such as a hardware address.
- The Internet Control Message Protocol (ICMP) is responsible for providing diagnostic functions and reporting errors or conditions regarding the delivery of IP packets.
- The Internet Group Management Protocol (IGMP) is responsible for the management of IP multicast groups.

The Internet Layer is analogous to the Network layer of the OSI model.

2.1.2Transport Layer

The Transport Layer (also known as the Host-to-Host Transport Layer) is responsible for providing the Application Layer with session and datagram communication services. The core protocols of the Transport Layer are TCP and the User Datagram Protocol (UDP).

- TCP provides a one-to-one, connection-oriented, reliable communications service. TCP is responsible for the establishment of a TCP connection, the sequencing and acknowledgment of packets sent, and the recovery of packets lost during transmission.
- UDP provides a one-to-one or one-to-many, connectionless, unreliable communications service. UDP is used when the amount of data to be transferred is small (such as the data that would fit into a single packet), when the overhead of establishing a TCP connection is not desired, or when the applications or upper layer protocols provide reliable delivery.

The Transport Layer encompasses the responsibilities of the OSI Transport Layer and some of the responsibilities of the OSI Session Layer.

2.1.3. Application Layer

The Application Layer provides applications the ability to access the services of the other layers and defines the protocols that applications use to exchange data. There are many Application Layer protocols and new protocols are always being developed.

The most widely known Application Layer protocols are those used for the exchange of user information:

- The HyperText Transfer Protocol (HTTP) is used to transfer files that make up the Web pages of the World Wide Web.
- The File Transfer Protocol (FTP) is used for interactive file transfer.
- The Simple Mail Transfer Protocol (SMTP) is used for the transfer of mail messages and attachments.
- Telnet, a terminal emulation protocol, is used for remote login to network hosts.

Additionally, the following Application Layer protocols help facilitates the use and management of TCP/IP networks:

- The Domain Name System (DNS) is used to resolve a host name to an IP address.
- The Routing Information Protocol (RIP) is a routing protocol that routers use to exchange routing information on an IP internetwork.
- The Simple Network Management Protocol (SNMP) is used between network management console and network devices (routers, bridges, and intelligent hubs) to collect and exchange network management information.

Examples of Application Layer interfaces for TCP/IP applications are Windows Sockets and NetBIOS. Windows Sockets provides a standard application-programming interface (API) under the Microsoft Windows operating system. NetBIOS is an industry-standard interface for accessing protocol services such as sessions, datagrams, and name resolution. More information on Windows Sockets and NetBIOS is provided later in this paper.

2.2. TCP/IP Core Protocols

The TCP/IP protocol component that is installed in your network operating system is a series of interconnected protocols called the core protocols of TCP/IP. All other applications and other protocols in the TCP/IP protocol suite rely on the basic services provided by the following protocols: IP, ARP, ICMP, IGMP, TCP, and UDP.

2.2.1. IP

IP is a connectionless, unreliable datagram protocol primarily responsible for addressing and routing packets between hosts. Connectionless means that a session is not established before exchanging data. Unreliable means that delivery is not guaranteed. IP will always make 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. IP is defined in RFC 791.

An IP packet consists of an IP header and an IP payload. Table 3 describes the key fields in the IP header.

Table 3 Key fields in the IP header

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.
Identification	Used to identify a specific IP datagram and to identify all fragments of a specific IP datagram if fragmentation occurs.
Protocol	Informs IP at the destination host whether to pass the packet up to TCP, UDP, ICMP, or other protocols.
Checksum	A simple mathematical computation used to verify the integrity of the IP header.
Time to Live (TTL)	Designates the number of networks 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 one.

Fragmentation and Reassembly

If a router receives an IP packet that is too large for the network onto which the packet is being forwarded, IP will fragment the original packet into smaller packets that will fit on the downstream network. When the packets arrive at their final destination, IP at the destination host reassembles the fragments into the original payload. This process is referred to as fragmentation and reassembly. Fragmentation can occur in environments that have a mix of networking technologies, such as Ethernet and Token Ring.

The fragmentation and reassembly works as follows:

1. When an IP packet is sent by the source, it places a unique value in the Identification field.
2. The IP packet is received at the router. The IP router notes that the maximum transmission unit (MTU) of the network onto which the packet is to be forwarded is smaller than the size of the IP packet.
3. IP fragments the original IP payload into fragments that will fit on the next network. Each fragment is sent with its own IP header which contains:
   - The original Identification field identifies all fragments that belong together.
   - The More Fragments Flag indicates that other fragments follow. The More Fragments Flag is not set on the last fragment, because no other fragments follow it.
   - The Fragment Offset field indicates the position of the fragment relative to the original IP payload.
4. When the fragments are received by IP at the remote host, they are identified by the Identification field as belonging together. The Fragment Offset is then used to reassemble the fragments into the original IP payload.

2.2.2. ARP

When IP packets are sent on shared access, broadcast-based networking technologies such as Ethernet or Token Ring, the Media Access Control (MAC) address corresponding to a forwarding IP address must be resolved. ARP uses MAC-level broadcasts to resolve a known forwarding IP address to its MAC address. ARP is defined in RFC 826.

more information on ARP, see the "Physical Address Resolution" section later in this paper.

2.2.3. ICMP

Internet Control Message Protocol (ICMP) provides troubleshooting facilities and error reporting for packets that are undeliverable. For example, if IP is unable to deliver a packet to the destination host, ICMP will send a Destination Unreachable message to the source host. Table 4 shows the most common ICMP messages.

Table 4 Common ICMP messages

ICMP Message	Function
Echo	Simple troubleshooting message used to check IP connectivity to a desired host.
Echo Reply	Response to an ICMP Echo.
Redirect	Sent by a router to inform a sending host of a better route to a destination IP address.
Source Quench	Sent by a router to inform a sending host that its IP datagrams are being dropped due to congestion at the router. The sending host then lowers its transmission rate. Source Quench is an elective ICMP message and is not commonly implemented.
Destination Unreachable	Sent by a router or the destination host to inform the sending host that the datagram cannot be delivered.

To send ICMP Echo messages and view statistics on the responses on a Windows 2000-based computer, use the ping utility at a Windows 2000 command prompt.

There are a series of defined Destination Unreachable ICMP messages. Table 5 describes the most common ICMP Destination Unreachable messages.

Table 5 Common ICMP Destination Unreachable messages

Destination Unreachable Message	**Description**
Host Unreachable	Sent by an IP router when a route to the destination network cannot be found.
Protocol Unreachable	Sent by the destination IP node when the Protocol field in the IP header cannot be matched with an IP client protocol currently loaded.
Port Unreachable	Sent by the destination IP node when the Destination Port in the UDP header cannot be matched with a process using that port.
Fragmentation Needed and DF Set	Sent by an IP router when fragmentation must occur but is not allowed due to the source node setting the Don't Fragment (DF) flag in the IP header.

ICMP does not make IP a reliable protocol. ICMP attempts to report errors and provide feedback on specific conditions. ICMP messages are carried as unacknowledged IP datagrams and are themselves unreliable. ICMP is defined in RFC 792.

2.2.4 IGMP

Internet Group Management Protocol (IGMP) is a protocol that manages host membership in IP multicast groups. An IP multicast group, also known as a host group, is a set of hosts that listen for IP traffic destined for a specific multicast IP address. Multicast IP traffic is sent to a single MAC address but processed by multiple IP hosts. A given host listens on a specific IP multicast address and receives all packets to that IP address. Some additional aspects of IP multicasting:

- Host group membership is dynamic; hosts can join and leave the group at any time.
- A host group can be of any size.
- Members of a host group can span IP routers across multiple networks. This situation requires IP multicast support on the IP routers and the ability for hosts to register their group membership with local routers. Host registration is accomplished using IGMP.
- A host can send traffic to an IP multicast address without belonging to the corresponding host group.

For a host to receive IP multicasts, an application must inform IP that it will be receiving multicasts at a specified destination IP multicast address. If the network technology supports hardware-based multicasting, then the network interface is told to pass up packets for a specific multicast address. In the case of Ethernet, the network interface card is programmed to respond to a multicast MAC address corresponding to the desired IP multicast address.

A host supports IP multicast at one of the following levels:

- Level 0

  No support to send or receive IP multicast traffic.

- Level 1

  Support exists to send but not receive IP multicast traffic.

- Level 2

  Support exists to both send and receive IP multicast traffic. TCP/IP for Windows 2000 supports level 2 IP multicasting.

The protocol to register host group information is IGMP. IGMP is required on all hosts that support level 2 IP multicasting. IGMP packets are sent using an IP header.

IGMP messages take two forms:

1. When a host joins a host group, it sends an IGMP Host Membership Report message to the all-hosts IP multicast address (224.0.0.1) or to the desired multicast address declaring its membership in a specific host group by referencing the IP multicast address.
2. When a router polls a network to ensure there are members of a specific host group, it sends an IGMP Host Membership Query message to the all-hosts IP multicast address. If no responses to the poll are received after several polls, the router assumes no membership in that group for that network and stops advertising that group-network information to other routers.

For IP multicasting to span routers across an internetwork, multicast routing protocols are used by routers to communicate host group information so that each router supporting multicast forwarding is aware of which networks contain members of which host groups.

2.2.5.TCP

TCP is a reliable, connection-oriented delivery service. The data is transmitted in segments. Connection-oriented means that a connection must be established before hosts can exchange data. Reliability is achieved by assigning a sequence number to each segment transmitted. An acknowledgment is used to verify that the other host received the data. For each segment sent, the receiving host must return an acknowledgment (ACK) within a specified period for bytes received. If an ACK is not received, the data is retransmitted. TCP is defined in RFC 793.

TCP uses byte-stream communications, wherein data within the TCP segment is treated as a sequence of bytes with no record or field boundaries. Table 6 describes the key fields in the TCP header.

Table 6 **Key fields in the TCP header**

Field	**Function**
Source Port	TCP port of sending host.
Destination Port	TCP port of destination host.
Sequence Number	The sequence number of the first byte of data in the TCP segment.
Acknowledgment Number	The sequence number of the byte the sender expects to receive next from the other side of the connection.
Window	The current size of a TCP buffer on the host sending this TCP segment to store incoming segments.
TCP Checksum	Verifies the bit-level integrity of the TCP header and the TCP data.

2.2.5.1. TCP Ports

A TCP port provides a specific location for delivery of TCP segments. Port numbers below 1024 are well-known ports and are assigned by the Internet Assigned Numbers Authority (IANA). Table 7 lists a few well-known TCP ports.

Table 7 . Well-known TCP ports

TCP Port Number	Description
20	FTP (Data Channel)
21	FTP (Control Channel)
23	Telnet
80	HyperText Transfer Protocol (HTTP) used for the World Wide Web
139	NetBIOS session service

2.2.5.2. The TCP Three-Way Handshake

A TCP connection is initialized through a three-way handshake. The purpose of the three-way handshake is to synchronize the sequence number and acknowledgment numbers of both sides of the connection, exchange TCP Window sizes, and exchange other TCP options such as the maximum segment size or the use of large window sizes or TCP timestamps. The following steps outline the process:

1. The client sends a TCP segment to the server with an initial Sequence Number for the connection and a Window size indicating the size of a buffer on the client to store incoming segments from the server.
2. The server sends back a TCP segment containing its chosen initial Sequence Number, an acknowledgment of the client's Sequence Number, and a Window size indicating the size of a buffer on the server to store incoming segments from the client.
3. The client sends a TCP segment to the server containing an acknowledgement of the server's Sequence Number.

TCP uses a similar handshake process to end a connection. This guarantees that both hosts have finished transmitting and that all data was received.

2.2.6. UDP

UDP provides a connectionless datagram service that offers unreliable, best-effort delivery of data transmitted in messages. This means that the arrival of datagrams is not guaranteed; nor is the correct sequencing of delivered packets. UDP does not recover from lost data through retransmission. UDP is defined in RFC 768.

UDP is used by applications that do not require an acknowledgment of receipt of data and that typically transmit small amounts of data at one time. The NetBIOS name service, NetBIOS datagram service, and the Simple Network Management Protocol (SNMP) are examples of services and applications that use UDP. Table 8 describes the key fields in the UDP header.

Table 8 Key fields in the UDP header

Field	**Function**
Source Port	UDP port of sending host.
Destination Port	UDP port of destination host.
UDP Checksum	Verifies the bit-level integrity of the UDP header and the UDP data.
Acknowledgment Number	The sequence number of the byte the sender expects to receive next from the other side of the connection.

2.2.6.1. UDP Ports

To use UDP, an application must supply the IP address and UDP port number of the destination application. A port provides a location for sending messages. A port functions as a multiplexed message queue, meaning that it can receive multiple messages at a time. Each port is identified by a unique number. It's important to note that UDP ports are distinct and separate from TCP ports even though some of them use the same number. Table 9 lists well known UDP ports.

Table 9 **Well-known UDP ports**

UDP Port Number	Description
53	Domain Name System (DNS) Name Queries
69	Trivial File Transfer Protocol (TFTP)
137	NetBIOS name service
138	NetBIOS datagram service
161	Simple Network Management Protocol (SNMP)

161	Simple Network Management Protocol (SNMP)

For a complete list of assigned UDP ports, see http://www.isi.edu/in-notes/iana/assignments/port-numbers.

2.2.2. TCP/IP Application Interfaces

To allow applications to access the services offered by the core TCP/IP protocols in a standard way, network operating systems like Windows 2000 make industry standard application programming interfaces (APIs) available. Application programming interfaces are sets of functions and commands that are programmatically called by application code to perform network functions. For example, a Web browser application connecting to a Web site needs access to TCP's connection establishment service.

Figure 2 shows two common TCP/IP application interfaces, Windows Sockets, and NetBIOS, and their relation to the core protocols.

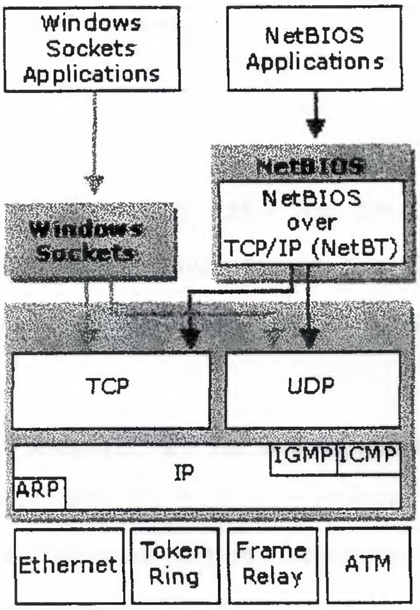


Figure 2 Application interfaces for TCP/IP

2.2.3. Windows Sockets Interface

The Windows Sockets API is a standard interface under Microsoft Windows for applications that use TCP and UDP. Applications written to the Windows Sockets API will run on many versions of TCP/IP. TCP/IP utilities and the Microsoft SNMP service are examples of applications written to the Windows Sockets interface.

Windows Sockets provides services that allow applications to bind to a particular port and IP address on a host, initiate and accept a connection, send and receive data, and close a connection. There are two types of sockets:

1. A stream socket provides a two-way, reliable, sequenced, and unduplicated flow of data using TCP.
2. A datagram socket provides the bi-directional flow of data using UDP.

A socket is defined by a protocol and an address on the host. The format of the address is specific to each protocol. In TCP/IP, the address is the combination of the IP address and port. Two sockets, one for each end of the connection, form a bi-directional communications path.

To communicate, an application specifies the protocol, the IP address of the destination host, and the port of the destination application. Once the application is connected, information can be sent and received.

2.2.4. NetBIOS Interface

NetBIOS (Network Basic Input/Output System) was developed for IBM in 1983 by Sytek Corporation to allow applications to communicate over a network. NetBIOS defines two entities, a session level interface and a session management/data transport protocol.

The NetBIOS interface is a standard API for user applications to submit network I/O and control directives to underlying network protocol software. An application program that uses the NetBIOS interface API for network communication can be run on any protocol software that supports the NetBIOS interface.

NetBIOS also defines a protocol that functions at the session/transport level. This is implemented by the underlying protocol software, such as the NetBIOS Frames Protocol (NBFP, a component of NetBEUI) or NetBIOS over TCP/IP (NetBT), to perform the network I/O required to accommodate the NetBIOS interface command set. NetBIOS over TCP/IP is defined in RFCs 1001 and 1002. NetBT is enabled by default. However, Windows 2000 allows you to disable NetBT for an environment that contains no NetBIOS-based network clients or applications.

NetBIOS provides commands and support for NetBIOS Name Management, NetBIOS Datagrams, and NetBIOS Sessions.

2.2.4.1. NetBIOS Name Management

NetBIOS Name Management services provide the following functions:

- Name Registration and Release
  When a TCP/IP host initializes, it registers its NetBIOS names by broadcasting or directing a NetBIOS name registration request to a NetBIOS Name Server such as a Windows Internet Name Service (WINS) server. If another host has registered the same NetBIOS name, either the host or a NetBIOS Name Server responds with a negative name

registration response. The initiating host receives an initialization error as a result.

When the workstation service on a host is stopped, the host discontinues broadcasting a negative name registration response when someone else tries to use the name and sends a name release to a NetBIOS Name Server. The NetBIOS name is said to be released and available for use by another host.

- Name Resolution
  When a NetBIOS application wants to communicate with another NetBIOS application, the IP address of the NetBIOS application must be resolved. NetBIOS over TCP/IP performs this function by either broadcasting a NetBIOS name query on the local network or sending a NetBIOS name query to a NetBIOS Name Server.

For more information on NetBIOS name resolution, see the "NetBIOS Name Resolution" section of this paper.

The NetBIOS Name Service uses UDP port 137.

2.2.4.2. NetBIOS Datagrams

The NetBIOS datagram service provides delivery of datagrams that are connectionless, non-sequenced, and unreliable. Datagrams can be directed to a specific NetBIOS name or broadcast to a group of names. Delivery is unreliable in that only the users who are logged on to the network will receive the message. The datagram service can initiate and receive both broadcast and directed messages. The NetBIOS Datagram service uses UDP port 138.

2.2.4.3. NetBIOS Sessions

The NetBIOS session service provides delivery of NetBIOS messages that are connection-oriented, sequenced, and reliable. NetBIOS sessions use TCP connections and provide session establishment, keep alive, and termination. The NetBIOS Session service allows concurrent data transfers in both directions using TCP port 139.

CHAPTER THREE: IP ADDRESSING

Each TCP/IP host is identified by a logical IP address. The IP address is a network layer address and has no dependence on the data link layer address (such as a MAC address of a network interface card). A unique IP address is required for each host and network component that communicates using TCP/IP.

The IP address identifies a system's location on the network in the same way a street address identifies a house on a city block. Just as a street address must identify a unique residence, an IP address must be globally unique and have a uniform format.

Each IP address includes a network ID and a host ID.

- The network ID (also known as a network address) identifies the systems that are located on the same physical network bounded by IP routers. All systems on the same physical network must have the same network ID. The network ID must be unique to the internetwork.
- The host ID (also known as a host address) identifies a workstation, server, router, or other TCP/IP host within a network. The address for each host must be unique to the network ID.

Note The use of the term network ID refers to any IP network ID, whether it is class-based, a subnet, or a supernet.

An IP address is 32 bits long. Rather than working with 32 bits at a time, it is a common practice to segment the 32 bits of the IP address into four 8-bit fields called octets. Each octet is converted to a decimal number (the Base 10 numbering system) in the range 0-255 and separated by a period (a dot). This format is called dotted decimal notation. Table 10 provides an example of an IP address in binary and dotted decimal formats.

Table 10 **Example of an IP address in binary and dotted decimal format**

Binary Format	Dotted Decimal Notation
11000000 10101000 00000011 00011000	192.168.3.24

The notation w.x.y.z is used when referring to a generalized IP address and shown in Figure 3.

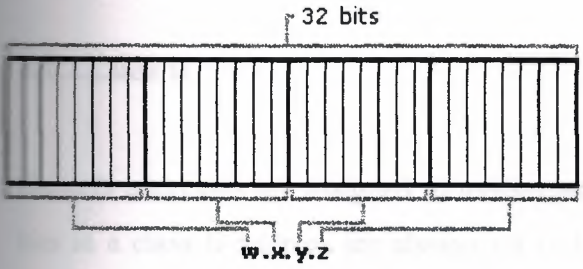


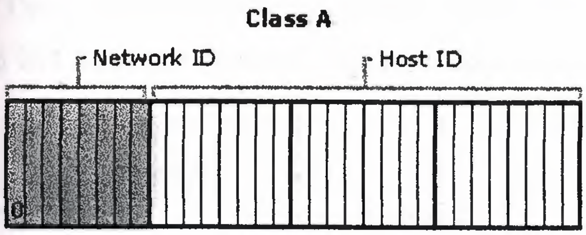
Figure 3 The IP address

3.1. Address Classes

The Internet community originally defined five address classes to accommodate networks of varying sizes. TCP/IP for Windows 2000 supports class A, B, and C addresses assigned to hosts. The class of address defines which bits are used for the network ID and which bits are used for the host ID. It also defines the possible number of networks and the number of hosts per network.

3.1.1. Class A

Class A addresses are assigned to networks with a very large number of hosts. The high-order bit in a class A address is always set to zero. The next seven bits (completing the first octet)

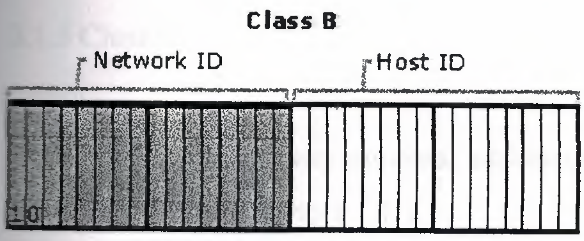


complete the network ID. The remaining 24 bits (the last three octets) represent the host ID. This allows for 126 networks and 16,777,214 hosts per network. Figure 4 illustrates the structure of class A addresses.

Figure 4 Class A IP addresses

3.1.2.Class B

Class B addresses are assigned to medium-sized to large-sized networks. The two high-order bits in a class B address are always set to binary 1 0. The next 14 bits (completing the first two octets) complete the network ID. The remaining 16 bits (last two octets)



represent the host ID. This allows for 16,384 networks and 65,534 hosts per network. Figure 5 illustrates the structure of class B addresses.

Figure 5 Class B IP addresses

3.1.3. Class C

Class C addresses are used for small networks. The three high-order bits in a class C address are always set to binary 1 1 0. The next 21 bits (completing the first three octets) complete the network ID. The remaining 8 bits (last octet) represent the host ID. This allows for 2,097,152 networks and 254 hosts per network. Figure 6 illustrates the structure of class C addresses.

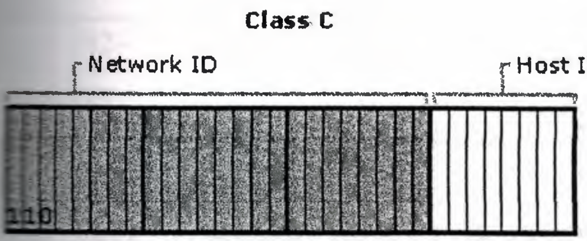


Figure 6 Class C IP addresses

3.1.4. Class D

Class D addresses are reserved for IP multicast addresses. The four high-order bits in a class D address are always set to binary 1 1 1 0. The remaining bits are for the address that interested hosts will recognize. Microsoft supports class D addresses for applications to multicast data to multicast-capable hosts on an internetwork.

3.1.5 Class E

Class E addresses are experimental addresses reserved for future use. The high-order bits in a class E address are set to 1 1 1 1.

Table 11 is a summary of address classes A, B, and C that can be used for host IP addresses.

Table 11 IP address class summary

Class	**Value for w[1]**	**Network ID Portion**	**Host ID Portion**	**Available Networks**	**Hosts per Network**
A	1–126	w	x.y.z	126	16,777,21

					4
B	128–191	w.x	y.z	16,384	65,534
C	192–223	w.x.y	z	2,097,152	254

* The class A address 127.x.y.z is reserved for loopback testing and interprocess communication on the local computer.

3.2. Network ID Guidelines

The network ID identifies the TCP/IP hosts that are located on the same physical network. All hosts on the same physical network must be assigned the same network ID to communicate with each other.

Follow these guidelines when assigning a network ID:

- The network address must be unique to the IP internetwork. If you plan on having a direct routed connection to the public Internet, the network ID must be unique to the Internet. If you do not plan on connecting to the public Internet, the local network ID must be unique to your private internetwork.
- The network ID cannot begin with the number 127. The number 127 in a class A address is reserved for internal loopback functions.
- All bits within the network ID cannot be set to 1. All 1's in the network ID are reserved for use as an IP broadcast address.
- All bits within the network ID cannot be set to 0. All 0's in the network ID are used to denote a specific host on the local network and will not be routed.

Table 12 lists the valid ranges of network IDs based on the IP address classes. To denote IP network IDs, the host bits are all set to 0. Note that even though expressed in dotted decimal notation, the network ID is not an IP address.

Table 12 Class ranges of network IDs

Address Class	First Network ID	Last Network ID
Class A	1.0.0.0	126.0.0.0
Class B	128.0.0.0	191.255.0.0
Class C	192.0.0.0	223.255.255.0

3.3. Host ID Guidelines

The host ID identifies a TCP/IP host within a network. The combination of IP network ID and IP host ID is an IP address.

Follow these guidelines when assigning a host ID:

- The host ID must be unique to the network ID.
- All bits within the host ID cannot be set to 1, because this host ID is reserved as a broadcast address to send a packet to all hosts on a network.
- All bits in the host ID cannot be set to 0, because this host ID is reserved to denote the IP network ID.

Table 13 lists the valid ranges of host IDs based on the IP address classes.

Table 13 Class ranges of host IDs

Address Class	First Host ID	Last Host ID
Class A	w.0.0.1	w.255.255.254
Class B	w.x.0.1	w.x.255.254
Class C	w.x.y.1	w.x.y.254

3.4 Subnets and Subnet Masks

The Internet Address Classes were designed to accommodate three different scales of IP internetworks, where the 32 bits of the IP address are apportioned between network IDs and host IDs depending on how many networks and hosts per network are needed. However, consider the class A network ID, which has the possibility of over 16 million hosts on the same network. All the hosts on the same physical network bounded by IP routers share the same broadcast traffic; they are in the same broadcast domain. It is not practical to have 16 million nodes in the same broadcast domain. The result is that most of the 16 million host addresses are not assignable and are wasted. Even a class B network with 65 thousand hosts is impractical.

In an effort to create smaller broadcast domains and to better utilize the bits in the host ID, an IP network can be subdivided into smaller networks, each bounded by an IP router and assigned a new subnetted network ID, which is a subset of the original class-based network ID.

This creates subnets, subdivisions of an IP network each with their own unique subnetted network ID. Subnetted network IDs are created by using bits from the host ID portion of the original class-based network ID.

Consider the example in Figure 7. The class B network of 131.107.0.0 can have up to 65,534 nodes. This is far too many nodes, and in fact, the current network is becoming saturated with broadcast traffic. The subnetting of network 131.107.0.0 should be done in such a way so that it does not impact, nor require, the reconfiguration of the rest of the IP internetwork.

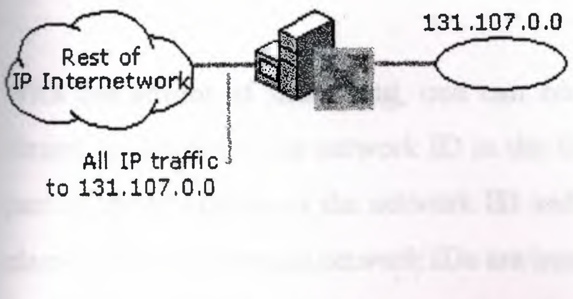


Figure 7   Network 131.107.0.0 before subnetting

Network 131.107.0.0 is subnetted by utilizing the first 8 host bits (the third octet) for the new subnetted network ID. When 131.107.0.0 is subnetted, as shown in Figure 8, separate networks with their own subnetted network IDs (131.107.1.0, 131.107.2.0, 131.107.3.0) are created. The router is aware of the separate subnetted network IDs and will route IP packets to the appropriate subnet.

Note that the rest of the IP internetwork still regards all the nodes on the three subnets as being on network 131.107.0.0. The other routers in the IP internetwork are unaware of the subnetting being done on network 131.107.0.0, and therefore require no reconfiguration.

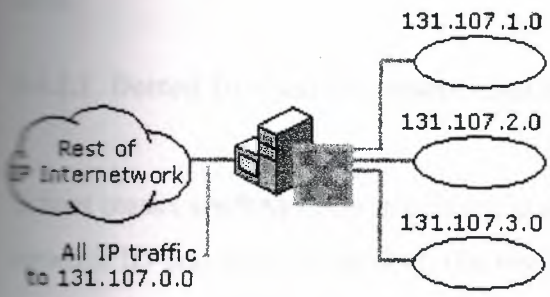


Figure 8 Network 131.107.0.0 after subnetting

A key element of subnetting is still missing. How does the router who is subdividing network 131.107.0.0 know how the network is being subdivided and which subnets are available on which router interfaces? To give the IP nodes this new level of awareness, it must be told exactly how to discern the new subnetted network ID regardless of Internet Address Classes. To tell an IP node exactly how to extract a network ID, either class-based or subnetted, a subnet mask is used.

3.4.1. Subnet Masks

With the advent of subnetting, one can no longer rely on the definition of the IP address classes to determine the network ID in the IP address. A new value is needed to define which part of the IP address is the network ID and which part is the host ID, regardless of whether class-based or subnetted network IDs are being used.

RFC 950 defines the use of a subnet mask (also referred to as an address mask) as a 32-bit value that is used to distinguish the network ID from the host ID in an arbitrary IP address. The bits of the subnet mask are defined as:

- All bits that correspond to the network ID are set to 1.
- All bits that correspond to the host ID are set to 0.

Each host on a TCP/IP network requires a subnet mask even on a single-segment network. Either a default subnet mask, which is used when using class-based network IDs, or a custom subnet mask, which is used when subnetting or supernetting, is configured on each TCP/IP node.

3.4.1.1. Dotted Decimal Representation of Subnet Masks

Subnet masks are frequently expressed in dotted decimal notation. Once the bits are set for the network ID and host ID portion, the resulting 32-bit number is converted to dotted decimal notation. Note that even though expressed in dotted decimal notation, a subnet mask is not an IP address.

A default subnet mask is based on the IP address classes and is used on TCP/IP networks that are not divided into subnets. Table 14 lists the default subnet masks using the dotted decimal notation for the subnet mask.

Table 14 Default subnet masks in dotted decimal notation

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

Custom subnet masks are those that differ from the above default subnet masks when doing subnetting or supernetting. For example, 131.107.58.0 is an 8-bit subnetted class B network ID. Eight bits of the class-based host ID are being used to express subnetted network IDs. The subnet mask uses a total of 24 bits (255.255.255.0) to define the subnetted network ID. The subnetted network ID and its corresponding subnet mask is then expressed in dotted decimal notation as:

131.107.58.0, 255.255.255.0

3.4.1.2. Network Prefix Length Representation of Subnet Masks

Since the network ID bits must be always chosen in a contiguous fashion from the high order bits, a shorthand way of expressing a subnet mask is to denote the number of bits that define the network ID as a network prefix using the network prefix notation: /<# of bits>. Table 15 lists the default subnet masks using the network prefix notation for the subnet mask.

Table 15 Default subnet masks in network prefix notation

Address Class	Bits for Subnet Mask	Network Prefix
Class A	11111111 00000000 00000000 00000000	/8
Class B	11111111 11111111 00000000 00000000	/16
Class C	11111111 11111111 11111111 00000000	/24

For example, the class B network ID 131.107.0.0 with the subnet mask of 255.255.0.0 would be expressed in network prefix notation as:

131.107.0.0/16.

As an example of a custom subnet mask, 131.107.58.0 is an 8-bit subnetted class B network ID. The subnet mask uses a total of 24 bits to define the subnetted network ID. The subnetted network ID and its corresponding subnet mask is then expressed in network prefix notation as:

131.107.58.0/24

Note Since all hosts on the same network must be using the same network ID, the ID must be defined by the same subnet mask. For example, 157.55.0.0/16 is not the same network ID as 157.55.0.0/24. The network ID 157.55.0.0/16 implies a range of valid host IP addresses from 157.55.0.1 to 157.55.255.254. The network ID 157.55.0.0/24 implies a range of valid host IP addresses from 157.55.0.1 to 157.55.0.254. Clearly, these network IDs do not represent the same range of IP addresses.

3.4.2. Determining the Network ID

To extract the network ID from an arbitrary IP address using an arbitrary subnet mask, IP uses a mathematical operation called a logical AND comparison. In an AND comparison, the result of two items being compared is true only when both items being compared are true, otherwise, the result is false. Applying this principle to bits, the result is 1 when both bits being compared are 1; otherwise, the result is 0.

IP takes the 32-bit IP address and logically ANDs it with the 32-bit subnet mask. This operation is known as a bit-wise logical AND. The result of the bit-wise logical AND of the IP address and the subnet mask is the network ID.

For example, what is the network ID of the IP node 131.107.189.41 with a subnet mask of 255.255.240.0?

To obtain the result, turn both numbers into their binary equivalents and line them up. Then perform the AND operation on each bit and write down the result.

10000011 01101011 10111101 00101001 IP Address

11111111 11111111 11110000 00000000 Subnet Mask

10000001 00111000 10110000 00000000 Network ID

The result of the bit-wise logical AND of the 32 bits of the IP address and the subnet mask is the network ID 131.107.176.0.

3.4.3. Subnetting

While the conceptual notion of subnetting by utilizing host bits are straightforward, the actual mechanics of subnetting are a bit more complicated. Subnetting is a three-step procedure:

1. Determine the number of host bits to be used for the subnetting.
2. Enumerate the new subnetted network IDs.
3. Enumerate the IP addresses for each new subnetted network ID.

Step 1: Determining the Number of Host Bits

The number of host bits being used for subnetting determines the possible number of subnets and hosts per subnet. Before you choose how many host bits, you should have a good idea of the number of subnets and hosts you will have in the future. Using more bits for the subnet mask than required will save you the time of reassigning IP addresses in the future.

The more host bits that are used, the more subnets (subnetted network IDs) you can have—but with fewer hosts. If you use too many host bits, it will allow for growth in the number of subnets, but will limit the growth in the number of hosts. If you use too few hosts, it will allow for growth in the number of hosts, but will limit the growth in the number of subnets.

For example, Figure 9 illustrates the subnetting of up to the first 8 host bits of a class B network ID. If we choose one host bit for subnetting, we obtain 2 subnetted network IDs with 16,382 hosts per subnetted network ID. If we choose 8 host bits for subnetting, we obtain 256 subnetted network IDs with 254 hosts per subnetted network ID.

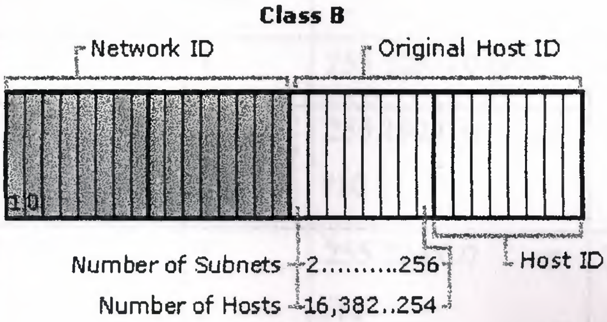


Figure 9 Subnetting a class B network ID

In practice, network administrators define a maximum number of nodes they want on a single network. Recall that all nodes on a single network share all the same broadcast traffic; they reside in the same broadcast domain. Therefore, growth in the amount of subnets is favored over growth in the amount of hosts per subnet.

Follow these guidelines to determine the number of host bits to use for subnetting.

1. Determine how many subnets you need now and will need in the future. Each physical network is a subnet. WAN connections may also count as subnets depending on whether your routers support unnumbered connections.
2. Use additional bits for the subnet mask if:
   - You will never require as many hosts per subnet as allowed by the remaining bits.
   - The number of subnets will increase in the future, requiring additional host bits.

To determine the desired subnetting scheme, you will start with an existing network ID to be subnetted. The network ID to be subnetted can be a class-based network ID, a subnetted network ID, or a supernet. The existing network ID will contain a series of network ID bits that are fixed, and a series of host ID bits which are variable. Based on your requirements for the number of subnets and the number of hosts per subnet, you will choose a specific number of host bits to be used for the subnetting.

Table 16 shows the subnetting of a class A network ID. Based on a required number of subnets, and a maximum number of hosts per subnet, a subnetting scheme can be chosen.

Table 16 Subnetting a Class A Network ID

Required number of subnets	**Number of host bits**	**Subnet Mask**	**Number of hosts per subnet**
1-2	1	255.128.0.0 or /9	8,388,606
3-4	2	255.192.0.0 or /10	4,194,302
5-8	3	255.224.0.0 or /11	2,097,150
9-16	4	255.240.0.0 or	1,048,574

		/12	
17-32	5	255.248.0.0 or /13	524,286
33-64	6	255.252.0.0 or /14	262,142
65-128	7	255.254.0.0 or /15	131,070
129-256	8	255.255.0.0 or /16	65,534
257-512	9	255.255.128.0 or /17	32,766
513-1,024	10	255.255.192.0 or /18	16,382
1,025-2,048	11	255.255.224.0 or /19	8,190
2,049-4,096	12	255.255.240.0 or /20	4,094
4,097-8,192	13	255.255.248.0 or /21	2,046
8,193-16,384	14	255.255.252.0 or /22	1,022
16,385-32,768	15	255.255.254.0 or /23	510
32,769-65,536	16	255.255.255.0 or /24	254
65,537-131,072	17	255.255.255.128 or /25	126
131,073-262,144	18	255.255.255.192 or /26	62

262,145-524,288	19	255.255.255.224 or /27	30
524,289-1,048,576	20	255.255.255.240 or /28	14
1,048,577-2,097,152	21	255.255.255.248 or /29	6
2,097,153-4,194,304	22	255.255.255.252 or /30	2

Table 17 shows the subnetting of a class B network ID.

Table 17 Subnetting a class B network ID

Required number of subnets	**Number of host bits**	**Subnet Mask**	**Number of hosts per subnet**
1-2	1	255.255.128.0 or /17	32,766
3-4	2	255.255.192.0 or /18	16,382
5-8	3	255.255.224.0 or /19	8,190
9-16	4	255.255.240.0 or /20	4,094
17-32	5	255.255.248.0 or /21	2,046
33-64	6	255.255.252.0 or /22	1,022
65-128	7	255.255.254.0 or /23	510

129-256	8	255.255.255.0 or /24	254
257-512	9	255.255.255.128 or /25	126
513-1,024	10	255.255.255.192 or /26	62
1,025-2,048	11	255.255.255.224 or /27	30
2,049-4,096	12	255.255.255.240 or /28	14
4,097-8,192	13	255.255.255.248 or /29	6
8,193-16,384	14	255.255.255.252 or /30	2

Table 18 shows the subnetting of a class C network ID.

Table 18 Subnetting a class C network ID

Required number of subnets	**Number of host bits**	**Subnet Mask**	**Number of hosts per subnet**
1-2	1	255.255.255.128 or /25	126
3-4	2	255.255.255.192 or /26	62
5-8	3	255.255.255.224 or /27	30
9-16	4	255.255.255.240	14

		or /28	
17-32	5	255.255.255.248 or /29	6
33-64	6	255.255.255.252 or /30	2

The All-Zeros and All-Ones Subnets

RFC 950 originally forbade the use of the subnetted network IDs where the bits being used for subnetting are set to all 0's (the all-zeros subnet) and all 1's (the all-ones subnet). The all-zeros subnet caused problems for early routing protocols and the all-ones subnet conflicts with a special broadcast address called the all-subnets directed broadcast address.

However, RFC 1812 now permits the use of the all-zeros and all-ones subnets in a Classless Inter-Domain Routing (CIDR)-compliant environment. CIDR-compliant environments use modern routing protocols that do not have a problem with the all-zeros subnet and the use of the all-subnets directed broadcast has been deprecated.

Before you use the all-zeros and all-ones subnets, verify that your hosts and routers support them. Windows 2000 supports the use of the all-zeros and all-ones subnets.

Step 2: Enumerating Subnetted Network IDs

Based on the number of host bits you use for your subnetting, you must list the new subnetted network IDs. There are two main approaches:

- Binary—List all possible combinations of the host bits chosen for subnetting and convert each combination to dotted decimal notation.
- Decimal—Add a calculated increment value to each successive subnetted network ID and convert to dotted decimal notation.

Either method produces the same result—the enumerated list of subnetted network IDs.

Note There are a variety of documented shortcut techniques for subnetting. However, they only work under a specific set of constraints (for example, only up to 8 bits of a class-based network ID). The methods described below are designed to work for any subnetting situation (class-based, more than 8 bits, supernetting, variable length subnetting).

3.4.3.1. Binary Subnetting Procedure

1. Based on n, the number of host bits chosen for subnetting, create a 3-column table with 2^n entries. The first column is the subnet number (starting with 1), the second column is the binary representation of the subnetted network ID, and the third column is the dotted decimal representation of the subnetted network ID.

   For each binary representation, the bits of the network ID being subnetted are fixed to their appropriate values and the remaining host bits are set to all 0's. The host bits chosen for subnetting will vary.

2. In the first table entry, set the subnet bits to all 0's and convert to dotted decimal notation. The original network ID is subnetted with its new subnet mask.
3. In the next table entry, increase the value within the subnet bits.
4. Convert the binary result to dotted decimal notation.
5. Repeat steps 3 and 4 until the table is complete.

As an example, a 3-bit subnet of the private network ID 192.168.0.0 is needed. The subnet mask for the new subnetted network IDs is 255.255.224.0 or /19. Based on n = 3, construct a table with 8 (= 2^3) entries. The entry for subnet 1 is the all 0's subnet. Additional entries in the table are successive increments of the subnet bits as shown in Table 19. The host bits used for subnetting are underlined.

Table 19 Binary subnetting technique for network ID 192.168.0.0

Subnet	Binary Representation	Subnetted Network ID
1	11000000.10101000.00000000.00000000	192.168.0.0/19
2	11000000.10101000.00100000.00000000	192.168.32.0/19
3	11000000.10101000.01000000.00000000	192.168.64.0/19
4	11000000.10101000.01100000.00000000	192.168.96.0/19
5	11000000.10101000.10000000.00000000	192.168.128.0/19

6	11000000.10101000.10100000.00000000	192.168.160.0/19
7	11000000.10101000.11000000.00000000	192.168.192.0/19
8	11000000.10101000.11100000.00000000	192.168.224.0/19

3.4.3.2. Decimal Subnetting Procedure

1. Based on n, the number of host bits chosen for subnetting, create a 3-column table with 2^n entries. The first column is the subnet number (starting with 1), the second column is the decimal (Base 10 numbering system) representation of the 32-bit subnetted network ID, and the third column is the dotted decimal representation of the subnetted network ID.
2. Convert the network ID (w.x.y.z) being subnetted from dotted decimal notation to N, a decimal representation of the 32-bit network ID.

N = w*16777216 + x*65536 + y*256 + z

3. Compute the increment value I based on h, the number of host bits remaining.

$I = 2^h$

4. In the first table entry, the decimal representation of the subnetted network ID is N and the subnetted network ID will be w.x.y.z with its new subnet mask.
5. In the next table entry, add I to the previous table entry's decimal representation.
6. Convert the decimal representation of the subnetted network ID to dotted decimal notation (W.X.Y.Z) through the following formula (where s is the decimal representation of the subnetted network ID):

W = INT(s/16777216)

X = INT((s mod(16777216))/65536)

Y = INT((s mod(65536))/256)

Z = s mod(256)

INT( ) denotes integer division, mod( ) denotes the modulus, the remainder upon division.

7. Repeat steps 5 and 6 until the table is complete.

As an example, a 3-bit subnet of the private network ID 192.168.0.0 is needed. Based on n = 3, we construct a table with 8 entries. The entry for subnet 1 is the all 0's subnet. N, the decimal representation of 192.168.0.0, is 3232235520, the result of 192*16777216 + 168*65536. Since there are 13 host bits remaining, the increment I is 2^{13} = 8192. Additional entries in the table are successive increments of 8192 as shown in Table 20.

Table 20 Decimal subnetting technique for network ID 192.168.0.0

Subnet	Decimal Representation	Subnetted Network ID
1	3232235520	192.168.0.0/19
2	3232243712	192.168.32.0/19
3	3232251904	192.168.64.0/19
4	3232260096	192.168.96.0/19
5	3232268288	192.168.128.0/19
6	3232276480	192.168.160.0/19
7	3232284672	192.168.192.0/19
8	3232292864	192.168.224.0/19

Step 3: Enumerating IP Addresses for Each Subnetted Network ID

Based on the enumeration of the subnetted network IDs, you must now list the valid IP addresses for new subnetted network IDs. To list each IP address individually would be too tedious. Instead, we will enumerate the IP addresses for each subnetted network ID by defining the range of IP addresses (the first and the last) for each subnetted network ID. There are two main approaches:

- Binary—Write down the first and last IP address for each subnetted network ID and convert to dotted decimal notation.
- Decimal—Add values incrementally, corresponding to the first and last IP addresses for each subnetted network ID and convert to dotted decimal notation.

Either method produces the same result—the range of IP addresses for each subnetted network ID.

3.4.4. Binary Procedure

1. Based on n, the number of host bits chosen for subnetting, create a 3-column table with 2^n entries. Alternately, add two columns to the previous table used for enumerating the subnetted network IDs. The first column is the subnet number (starting with 1), the second column is the binary representation of the first and last IP address for the subnetted network ID, and the third column is the dotted decimal representation of the first and last IP address of the subnetted network ID.
2. For each binary representation, the first IP address is the address where all the host bits are set to 0 except for the last host bit. The last IP address is the address where all the host bits are set to 1 except for the last host bit.
3. Convert the binary representation to dotted decimal notation.
4. Repeat steps 2 and 3 until the table is complete.

As an example, the range of IP addresses for the 3 bit subnetting of 192.168.0.0 is shown in Table 21. The bits used for subnetting are underlined.

Table 21 Binary enumeration of IP addresses

Subnet	Binary Representation	Range of IP Addresses
1	11000000.10101000.00000000.00000001 - 11000000.10101000.00011111.11111110	192.168.0.1 - 192.168.31.254
2	11000000.10101000.00100000.00000001 - 11000000.10101000.00111111.11111110	192.168.32.1 - 192.168.63.254
3	11000000.10101000.01000000.00000001 - 11000000.10101000.01011111.11111110	192.168.64.1 - 192.168.95.254
4	11000000.10101000.01100000.00	192.168.96.1 -

	000001 - 11000000.10101000.01111111.11 111110	192.168.127.254
5	11000000.10101000.10000000.00 000001 - 11000000.10101000.10011111.11 111110	192.168.128.1 - 192.168.159.254
6	11000000.10101000.10100000.00 000001 - 11000000.10101000.10111111.11 111110	192.168.160.1 - 192.168.191.254
7	11000000.10101000.11000000.00 000001 - 11000000.10101000.11011111.11 111110	192.168.192.1 - 192.168.223.254
8	11000000.10101000.11100000.00 000001 - 11000000.10101000.11111111.11 111110	192.168.224.1 - 192.168.255.254

3.4.5. Decimal Procedure

1. Based on n, the number of host bits chosen for subnetting, create a 3-column table with 2^n entries. Alternately, add two columns to the previous table used for enumerating the subnetted network IDs. The first column is the subnet number (starting with 1), the second column is the decimal representation of the first and last IP address for the subnetted network ID, and the third column is the dotted decimal representation of the first and last IP address of the subnetted network ID.
2. Compute the increment value J based on h, the number of host bits remaining.

$J = 2^h - 2$

3. For each decimal representation, the first IP address is N + 1 where N is the decimal representation of the subnetted network ID. The last IP address is N + J.

4. Convert the decimal representation of the first and last IP addresses to dotted decimal notation (W.X.Y.Z) through the following formula (where s is the decimal representation of the first or last IP address):

W = INT(s/16777216)

X = INT((s mod(16777216))/65536)

Y = INT((s mod(65536))/256)

Z = s mod(256)

INT( ) denotes integer division, mod( ) denotes the modulus, the remainder upon division.

5. Repeat steps 3 and 4 until the table is complete.

As an example, the range of IP addresses for the 3 bit subnetting of 192.168.0.0 is shown in Table 22. The increment J is 2^{13} - 2 = 8190.

Table 22 Decimal enumeration of IP addresses

Subnet	Decimal Representation	Range of IP Addresses
1	3232235521 – 3232243710	192.168.0.1 - 192.168.31.254
2	3232243713 – 3232251902	192.168.32.1 - 192.168.63.254
3	3232251905 – 3232260094	192.168.64.1 - 192.168.95.254
4	3232260097 – 3232268286	192.168.96.1 - 192.168.127.254
5	3232268289 – 3232276478	192.168.128.1 - 192.168.159.254
6	3232276481 – 3232284670	192.168.160.1 - 192.168.191.254
7	3232284673 – 3232292862	192.168.192.1 - 192.168.223.254
8	3232292865 – 3232301054	192.168.224.1 - 192.168.255.254

3.4.6. Variable Length Subnetting

One of the original uses for subnetting was to subdivide a class-based network ID into a series of equal-sized subnets. For example, a 4-bit subnetting of a class B network ID produced 16 equal-sized subnets (using the all-ones and all-zeros subnets). However, subnetting is a general method of utilizing host bits to express subnets and does not require equal-sized subnets.

Subnets of different size can exist within a class-based network ID. This is well-suited to real world environments, where networks of an organization contain different amounts of hosts, and different-sized subnets are needed to minimize the wasting of IP addresses. The creation and deployment of various-sized subnets of a network ID is known as variable length subnetting and uses variable length subnet masks (VLSM).

Variable length subnetting is a technique of allocating subnetted network IDs that use subnet masks of different sizes. However, all subnetted network IDs are unique and can be distinguished from each other by their corresponding subnet mask.

The mechanics of variable length subnetting are essentially that of performing subnetting on a previously subnetted network ID. When subnetting, the network ID bits are fixed and a certain amount of host bits are chosen to express subnets. With variable length subnetting, the network ID being subnetted has already been subnetted.

Variable Length Subnetting Example

For example, given the class-based network ID of 135.41.0.0/16, a required configuration is to reserve half of the addresses for future use, create 15 subnets with up to 2,000 hosts, and 8 subnets with up to 250 hosts.

3.4.7. Reserve half of the addresses for future use

To reserve half of the addresses for future use, a 1-bit subnetting of the class-based network ID of 135.41.0.0 is done, producing 2 subnets, 135.41.0.0/17 and 135.41.128.0/17. The subnet 135.41.0.0/17 is chosen as the portion of the addresses that are reserved for future use.

Table 23 shows one subnet with up to 32,766 hosts.

Table 23 **Reserving half the addresses for future use**

Subnet Number	**Network ID (dotted decimal)**	**Network ID (network prefix)**
1	135.41.0.0, 255.255.128.0	135.41.0.0/17

3.4.8. Fifteen Subnets with up to 2,000 Hosts

To achieve a requirement of 15 subnets with approximately 2,000 hosts, a 4-bit subnetting of the subnetted network ID of 135.41.128.0/17 is done. This produces 16 subnets (135.41.128.0/21, 135.41.136.0/21 . . . 135.41.240.0/21, 135.41.248.0/21), allowing up to 2,046 hosts per subnet. The first 15 subnetted network IDs (135.41.128.0/21 to 135.41.240.0/21) are chosen as the network IDs, which fulfills the requirement.

Table 24 illustrates 15 subnets with up to 2,000 hosts.

Table 24 15 Subnets with up to 2,046 hosts

Subnet Number	**Network ID (dotted decimal)**	**Network ID (network prefix)**
1	135.41.128.0, 255.255.248.0	135.41.128.0/21
2	135.41.136.0, 255.255.248.0	135.41.136.0/21
3	135.41.144.0, 255.255.248.0	135.41.144.0/21
4	135.41.152.0, 255.255.248.0	135.41.152.0/21
5	135.41.160.0, 255.255.248.0	135.41.160.0/21
6	135.41.168.0, 255.255.248.0	135.41.168.0/21
7	135.41.176.0,	135.41.176.0/21

	255.255.248.0	
8	135.41.184.0, 255.255.248.0	135.41.184.0/21
9	135.41.192.0, 255.255.248.0	135.41.192.0/21
10	135.41.200.0, 255.255.248.0	135.41.200.0/21
11	135.41.208.0, 255.255.248.0	135.41.208.0/21
12	135.41.216.0, 255.255.248.0	135.41.216.0/21
13	135.41.224.0, 255.255.248.0	135.41.224.0/21
14	135.41.232.0, 255.255.248.0	135.41.232.0/21
15	135.41.240.0, 255.255.248.0	135.41.240.0/21

3.4.9. 8 Subnets with up to 250 Hosts

To achieve a requirement of 8 subnets with up to 250 hosts, a 3-bit subnetting of subnetted network ID of 135.41.248.0/21 is done, producing 8 subnets (135.41.248.0/24, 135.41.249.0/24 . . . 135.41.254.0/24, 135.41.255.0/24) and allowing up to 254 hosts per subnet. All 8 subnetted network IDs (135.41.248.0/24 to 135.41.255.0/24) are chosen as the network IDs, which fulfills the requirement.

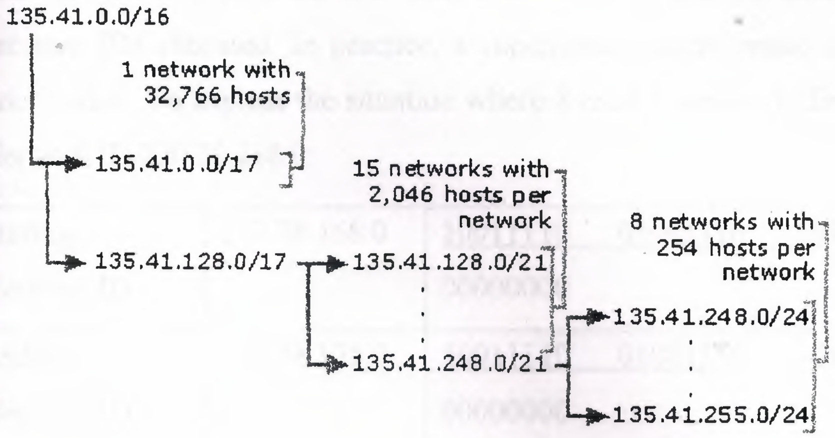
Table 25 illustrates 8 subnets with approximately 250 hosts.

Table 25 8 subnets with up to 254 hosts

Subnet Number	Network ID (dotted decimal)	Network ID (network prefix)
1	135.41.248.0, 255.255.255.0	135.41.248.0/24
2	135.41.249.0, 255.255.255.0	135.41.249.0/24
3	135.41.250.0, 255.255.255.0	135.41.250.0/24
4	135.41.251.0, 255.255.255.0	135.41.251.0/24
5	135.41.252.0, 255.255.255.0	135.41.252.0/24
6	135.41.253.0, 255.255.255.0	135.41.253.0/24
7	135.41.254.0, 255.255.255.0	135.41.254.0/24
8	135.41.255.0, 255.255.255.0	135.41.255.0/24

The variable length subnetting of 135.41.0.0/16 is shown graphically in Figure 10.

Figure 10 Variable length subnetting of 135.41.0.0/16


135.41.0.0/16
1 network with 32,766 hosts
135.41.0.0/17
135.41.128.0/17
15 networks with 2,046 hosts per network
135.41.128.0/21
135.41.248.0/21
8 networks with 254 hosts per network
135.41.248.0/24
135.41.255.0/24

Note In dynamic routing environments, variable length subnetting can only been deployed where the subnet mask is advertised along with the network ID. Routing Information Protocol (RIP) for IP version 1 does not support variable length subnetting. RIP for IP version 2, Open Short Path First (OSPF), and BGPv4 all support variable length subnetting.

3.4.10. Supernetting and Classless Inter-Domain Routing

With the recent growth of the Internet, it became clear to the Internet authorities that the class B network IDs would soon be depleted. For most organizations, a class C network ID does not contain enough host IDs and a class B network ID has enough bits to provide a flexible subnetting scheme within the organization.

The Internet authorities devised a new method of assigning network IDs to prevent the depletion of class B network IDs. Rather than assigning a class B network ID, the Internet Network Information Center (InterNIC) assigns a range of class C network IDs that contain enough network and host IDs for the organization's needs. This is known as supernetting. For example, rather than allocating a class B network ID to an organization that has up to 2,000 hosts, the InterNIC allocates a range of 8 class C network IDs. Each class C network ID accommodates 254 hosts, for a total of 2,032 host IDs.

While this technique helps conserve class B network IDs, it creates a new problem. Using conventional routing techniques, the routers on the Internet now must have 8 class C network ID entries in their routing tables to route IP packets to the organization. To prevent Internet routers from becoming overwhelmed with routes, a technique called Classless Inter-Domain Routing (CIDR) is used to collapse multiple network ID entries into a single entry corresponding to all of the class C network IDs allocated to that organization.

Conceptually, CIDR creates the routing table entry: {Starting Network ID, count}, where Starting Network ID is the first class C network ID and the count is the number of class C network IDs allocated. In practice, a supernetted subnet mask is used to convey the same information. To express the situation where 8 class C network IDs are allocated starting with Network ID 220.78.168.0:

Starting Network ID	220.78.168.0	10011110 01001110 10101000 00000000
Ending Network ID	220.78.175.0	10011110 01001110 10101111 00000000

Note that the first 21 bits (underlined) of all the above Class C network IDs are the same. The last three bits of the third octet vary from 000 to 111. The CIDR entry in the routing tables of the Internet routers becomes:

Network ID	Subnet Mask	Subnet Mask (binary)
220.78.168.0	255.255.248.0	11111111 11111111 11111000 00000000

In network prefix notation, the CIDR entry is 220.78.168.0/21.

A block of addresses using CIDR is known as a CIDR block.

Note Since subnet masks are used to express the count, class-based network IDs must be allocated in groups corresponding to powers of two.

In order to support CIDR, routers must be able to exchange routing information in the form of {Network ID, Subnet Mask} pairs. RIP for IP version 2, OSPF, and BGPv4 are routing protocols that support CIDR. RIP for IP version 1 does not support CIDR.

3.4.11. The Address Space Perspective

The use of CIDR to allocate addresses promotes a new perspective on IP network IDs. In the above example, the CIDR block {220.78.168.0, 255.255.248.0} can be thought of in two ways:

- A block of 8 class C network IDs.
- An address space in which 21 bits are fixed and 11 bits are assignable.

In the latter perspective, IP network IDs lose their class-based heritage and become separate IP address spaces, subsets of the original IP address space defined by the 32-bit IP address. Each IP network ID (class-based, subnetted, CIDR block), is an address space in which certain bits are fixed (the network ID bits) and certain bits are variable (the host bits). The host bits are assignable as host IDs or, using subnetting techniques, can be used in whatever manner best suits the needs of the organization.

3.4.12. Public and Private Addresses

If your intranet is not connected to the Internet, any IP addressing can be deployed. If direct (routed) or indirect (proxy or translator) connectivity to the Internet is desired, then there are two types of addresses employed on the Internet, public addresses and private addresses.

3.4.12.1. Public Addresses

Public addresses are assigned by InterNIC and consist of class-based network IDs or blocks of CIDR-based addresses (called CIDR blocks) that are guaranteed to be globally unique to the Internet.

When the public addresses are assigned, routes are programmed into the routers of the Internet so that traffic to the assigned public addresses can reach their locations. Traffic to destination public addresses are reachable on the Internet.

For example, when an organization is assigned a CIDR block in the form of a network ID and subnet mask, that {network ID, subnet mask} pair also exists as a route in the routers of the Internet. IP packets destined to an address within the CIDR block are routed to the proper destination.

3.4.12.2. Illegal Addresses

Private intranets that have no intent on connecting to the Internet can choose any addresses they want, even public addresses that have been assigned by the InterNIC. If an organization later decides to connect to the Internet, its current address scheme may include addresses already assigned by the InterNIC to other organizations. These addresses would be duplicate or conflicting addresses and are known as illegal addresses. Connectivity from illegal addresses to Internet locations is not possible.

For example, a private organization chooses to use 207.46.130.0/24 as its intranet address space. The public address space 207.46.130.0/24 has been assigned to the Microsoft corporation and routes exist on the Internet routers to route all packets destined to IP addresses on 207.46.130.0/24 to Microsoft routers. As long as the private organization does not connect to the Internet, there is no problem, since the two address spaces are on separate IP internetworks. If the private organization then connected directly to the Internet and continued to use 207.46.130.0/24 as its address space, then any Internet response traffic to locations on the 207.46.130.0/24 network would be routed to Microsoft routers, not to the routers of the private organization.

3.4.12.3. Private Addresses

Each IP node requires an IP address that is globally unique to the IP internetwork. In the case of the Internet, each IP node on a network connected to the Internet requires an IP address that is globally unique to the Internet. As the Internet grew, organizations connecting to the

Internet required a public address for each node on their intranets. This requirement placed a huge demand on the pool of available public addresses.

When analyzing the addressing needs of organizations, the designers of the Internet noted that for many organizations, most of the hosts on the organization's intranet did not require direct connectivity to Internet hosts. Those hosts that did require a specific set of Internet services, such as the World Wide Web access and e-mail, typically access the Internet services through application layer gateways such as proxy servers and e-mail servers. The result is that most organizations only required a small amount of public addresses for those nodes (such as proxies, routers, firewalls, and translators) that were directly connected to the Internet.

For the hosts within the organization that do not require direct access to the Internet, IP addresses that do not duplicate already-assigned public addresses are required. To solve this addressing problem, the Internet designers reserved a portion of the IP address space and named this space the private address space. An IP address in the private address space is never assigned as a public address. IP addresses within the private address space are known as private addresses. Because the public and private address spaces do not overlap, private addresses never duplicate public addresses.

The private address space specified in RFC 1597 is defined by the following three address blocks:

- 10.0.0.0/8

  The 10.0.0.0/8 private network is a class A network ID that allows the following range of valid IP addresses: 10.0.0.1 to 10.255.255.254. The 10.0.0.0/8 private network has 24 host bits that can be used for any subnetting scheme within the private organization.

- 172.16.0.0/12

  The 172.16.0.0/12 private network can be interpreted either as a block of 16 class B network IDs or as a 20-bit assignable address space (20 host bits) which can be used for any subnetting scheme within the private organization. The 172.16.0.0/12 private network allows the following range of valid IP addresses: 172.16.0.1 to 172.31.255.254.

- 192.168.0.0/16

  The 192.168.0.0/16 private network can be interpreted either as a block of 256 class C network IDs or as a 16-bit assignable address space (16 host bits), which can be used for any subnetting scheme within the private organization. The 192.168.0.0/16 private network allows the following range of valid IP addresses: 192.168.0.1 to 192.168.255.254.

The result of many organizations using private addresses is that the private address space is re-used, helping to prevent the depletion of public addresses.

Since the IP addresses in the private address space will never be,assigned by the InterNIC as public addresses, there will never exist routes in the Internet routers for private addresses. Traffic to destination private addresses are not reachable on the Internet. Therefore, Internet traffic from a host that has a private address must either send its requests to an application layer gateway (such as a proxy server), which has a valid public address, or have its private address translated into a valid public address by a network address translator (NAT) before it is sent on the Internet.

CHAPTER FOUR:NAME RESOLUTION

While 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 wish to communicate. People are much better at using and remembering names than IP addresses.

If a name is used as an alias for the IP address, there must exist a mechanism for assigning names to IP nodes to ensure its uniqueness and resolving a name to its IP address.

In this section, we will discuss the mechanisms used for assigning and resolving host names (which are used by Windows Sockets applications), and NetBIOS names (which are used by NetBIOS applications).

4.1. Host Name Resolution

A host name is an alias assigned to an IP node to identify it as a TCP/IP host. The host name can be up to 255 characters long and can contain alphabetic and numeric characters and the "-" and "." characters. Multiple host names can be assigned to the same host.

Windows Sockets applications, such as Microsoft Internet Explorer and the FTP utility, can use one of two values for the destination to be connected—the IP address or a host name. When the IP address is specified, name resolution is not needed. When a host name is specified, the host name must be resolved to an IP address before IP-based communication with the desired resource can begin.

Host names can take various forms. The two most common forms are a nickname and a domain name. A nickname is an alias to an IP address that individual people can assign and use. A domain name is a structured name that follows Internet conventions.

4.2. Domain Names

To facilitate a variety of different types of organizations and their desires to have a scaleable, customizable naming scheme in which to operate, the InterNIC has created and maintains a hierarchical namespace called the Domain Name System (DNS). DNS is a naming scheme that looks similar to the directory structure for files on a disk. However, instead of tracing a file from the root directory through subdirectories to its final location and its file name, a host name is traced from its final location through its parent domains back up to the root. The unique name of the host, representing its position in the hierarchy, is called its Fully Qualified Domain Name (FQDN). The top-level domain namespace is shown in Figure 11 with example second level and subdomains.

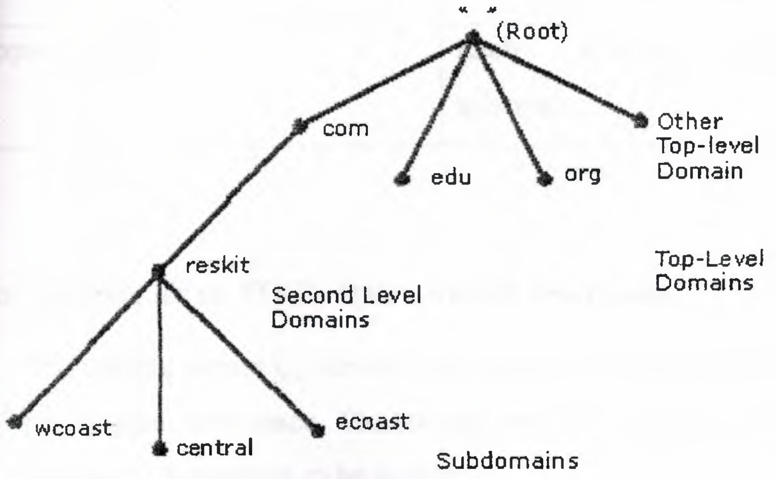


Figure 11 The Domain Name System

The parts of the domain namespace are:

- The root domain represents the root of the namespace and is indicated with a "" (null).
- Top-level domains, those directly below the root, indicate a type of organization. On the Internet, the InterNIC is responsible for the maintenance of top-level domain names. Table 26 has a partial list of the Internet's top-level domain names.
- Below the top-level domains are second-level domains, which identify a specific organization within its top-level domain. On the Internet, the InterNIC is responsible for the maintenance of second-level domain names and ensuring their uniqueness.

- Below the second-level domain are the subdomains of the organization. The individual organization is responsible for the creation and maintenance of subdomains.

Table 26 Internet top-level domain names

Domain Name	**Meaning**
COM	Commercial organization
EDU	Educational institution
GOV	Government institution
MIL	Military group
NET	Major network support center
ORG	Organization other than those above
INT	International organization
<country code>	Each country (geographic scheme)

For example, for the FQDN **ftpsrv.wcoast.reskit.com.**:

- The trailing period (.) denotes that this is an FQDN with the name relative to the root of the domain namespace. The trailing period is usually not required for FQDNs and if it is missing it is assumed to be present.
- **com** is the top-level domain, indicating a commercial organization.
- **reskit** is the second-level domain, indicating the Windows 2000 Resource Kit.
- **wcoast** is a subdomain of reskit.com indicating the West Coast division of the Windows 2000 Resource Kit.
- **ftpsrv** is the name of the FTP server in the West Coast division.

Domain names are not case sensitive. Organizations not connected to the Internet can implement whatever top and second-level domain names they want. However, typical implementations do adhere to the InterNIC specification so that an eventual participation in the Internet will not require a renaming process.

When a Windows Sockets application is given an FQDN as the destination location, the application calls a Windows Sockets function to resolve the name to an IP address. The request is passed to the DNS resolver component in the TCP/IP protocol. The DNS resolver packages the FQDN request as a DNS Name Query packet and sends it to the DNS server.

DNS is a distributed naming system. Rather than storing all the records for the entire namespace on each DNS server, each DNS server only stores the records for a specific portion of the namespace. The DNS server is authoritative for the portion of the namespace that corresponds to records stored on that DNS server. In the case of the Internet, hundreds of DNS servers store various portions of the Internet namespace. To facilitate the resolution of any valid domain name by any DNS server, DNS servers are also configured with pointer records to other DNS servers.

The following process outlines what happens when the DNS resolver component on a host sends a DNS query to a DNS server. This example is shown in Figure 12 and is deliberately simplified to gain a basic understanding of the DNS resolution process.

1. The DNS resolver component formats a DNS Name Query containing the FQDN and sends it to the configured DNS server.
2. The DNS server checks the FQDN in the DNS Name Query against locally stored address records. If a record is found, the IP address corresponding to the requested FQDN is sent back to the client.
3. If the FQDN is not found, the DNS server forwards the request to a DNS server that is authoritative for the FQDN.
4. The authoritative DNS server returns the reply, containing the resolved IP address, back to the original DNS server.

The original DNS server sends the IP address mapping information to the client.

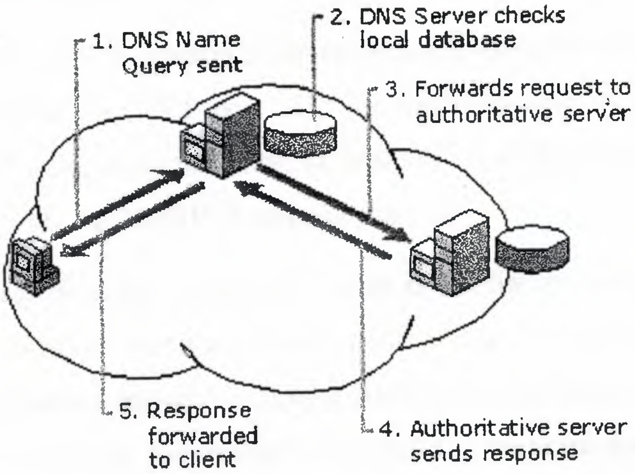

1. DNS Name Query sent
2. DNS Server checks local database
3. Forwards request to authoritative server
4. Authoritative server sends response
5. Response forwarded to client

Figure 12 An example of resolving an FQDN using DNS servers

To obtain the IP address of a server that is authoritative for the FQDN, DNS servers on the Internet go through an iterative process of querying multiple DNS servers until the authoritative server is found. More details on this iterative process can be found in the Windows 2000 Domain Name System Overview whitepaper.

4.5. Combining a Local Database File with DNS

TCP/IP implementations, including Windows 2000, allow the use of both a local database file and a DNS server to resolve host names. When a user specifies a host name in a TCP/IP command or utility, TCP/IP will:

1. Check the local database file (the HOSTS file) for a matching name.
2. If a matching name is not found in the local database file, the host name is packaged as a DNS Name Query and sent to the configured DNS server.

Combining both methods gives the user the abilities to have a local database file to resolve personalized nicknames and to use the globally distributed DNS database to resolve FQDNs.

4.6. NetBIOS Name Resolution

NetBIOS name resolution is the process of successfully mapping a NetBIOS name to an IP address. A NetBIOS name is a 16-byte address used to identify a NetBIOS resource on the network. A NetBIOS name is either a unique (exclusive) or group (non-exclusive) name. When a NetBIOS process is communicating with a specific process on a specific computer, a unique name is used. When a NetBIOS process is communicating with multiple processes on multiple computers, a group name is used.

The NetBIOS name acts as a session layer application identifier. For example, the NetBIOS Session service operates over TCP port 139. All NetBIOS over TCP/IP session requests will be addressed to TCP destination port 139. To identify which NetBIOS application to establish a NetBIOS session with, the NetBIOS name is used.

An example of a process using a NetBIOS name is the Server service on a Windows 2000–based computer. The Server service provides file and printer sharing. When your computer starts up, the Server service registers a unique NetBIOS name based on your computer's name. The exact name used by the Server service is the 15-character computer name plus a 16th character of 0x20. If the computer name is not 15 characters long, it is padded with spaces up to 15 characters long. Other network services also use the computer name to build their NetBIOS names so the 16th character is used to uniquely identify each service, such as the redirector, server, or messenger services. Figure 13 shows the NetBIOS names associated with the server, redirector, and messenger services.

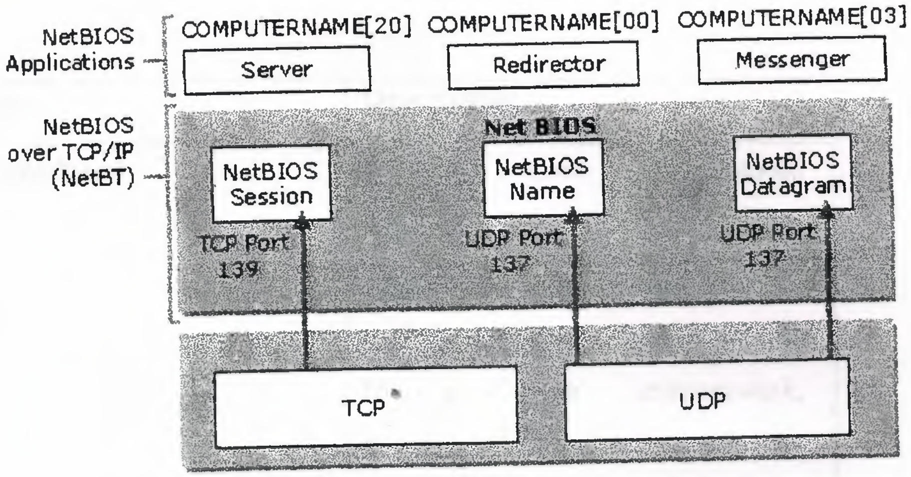


Figure 13   NetBIOS names and services

When you attempt to make a file sharing connection to a Windows 2000–based computer by name, the Server service on the file server you specify corresponds to a specific NetBIOS name. For example, when you attempt to connect to the computer called CORPSERVER, the NetBIOS name corresponding to the Server service on CORPSERVER is "**CORPSERVER <20>**" (note the padding using spaces). Before a file or printer sharing connection can be established, a TCP connection must be created. In order for a TCP connection to be established, the NetBIOS name "**CORPSERVER <20>**" must be resolved to an IP address.

To view the NetBIOS names registered by NetBIOS processes running on a Windows 2000-based computer, type “nbtstat –n” at the Windows 2000 command prompt.

4.7. NetBIOS Node Types

The exact mechanism by which NetBIOS names are resolved to IP addresses depends on the node’s configured NetBIOS Node Type. RFC 1001 define the NetBIOS Node Types, as listed in Table 27.

Table 27 NetBIOS Node Types

Node Type	**Description**
B-node (broadcast)	B-node uses broadcasted NetBIOS Name Queries for name registration and resolution. B-node has two major problems: (1) In a large internetwork, broadcasts can increase the network load, and (2) Routers typically do not forward broadcasts, so only NetBIOS names on the local network can be resolved.
P-node (peer-peer)	P-node uses a NetBIOS name server (NBNS), such as Windows Internet Name Service (WINS), to

	resolve NetBIOS names. P-node does not use broadcasts; instead, it queries the name server directly. The most significant problems with P-node are that all computers must be configured with the IP address of the NBNS, and if the NBNS is down, computers will not be able to communicate even on the local network.
M-node (mixed)	M-node is a combination of B-node and P-node. By default, an M-node functions as a B-node. If it is unable to resolve a name by broadcast, it uses the NBNS of P-node.
H-node (hybrid)	H-node is a combination of P-node and B-node. By default, an H-node functions as a P-node. If it is unable to resolve a name through the NetBIOS name server, it uses a broadcast to resolve the name.

When NetBIOS over TCP/IP is enabled, Windows 2000–based computers are B-node by default and become H-node when configured for a WINS server. Windows 2000 also utilizes a local database file called LMHOSTS to resolve remote NetBIOS names.

For more information on WINS, see the Windows 2000 Server Windows Internet Naming Service (WINS) Overview whitepaper.

CHAPTER FİVE: IP ROUTING

Once the host name or NetBIOS name is resolved to an IP address, the IP packet must be sent by the sending host to the resolved IP address. Routing is the process of forwarding a packet based on the destination IP address. Routing occurs at a sending TCP/IP host and at an IP router. A router is a device that forwards the packets from one network to another. Routers are also commonly referred to as gateways. In both cases, sending host and router, a decision has to be made about where the packet is forwarded.

To make these decisions, the IP layer consults a routing table that is stored in memory. Routing table entries are created by default when TCP/IP initializes and additional entries are added either manually by a system administrator, or automatically through communication with routers.

5.1. Direct and Indirect Delivery

Forwarded IP packets use at least one of two types of delivery based on whether the IP packet is forwarded to the final destination or whether it is forwarded to an IP router. These two types of delivery are known as direct and indirect delivery.

Direct delivery occurs when the IP node (either the sending node or an IP router) forwards a packet to the final destination on a directly attached network. The IP node encapsulates the IP datagram in a frame format for the Network Interface Layer (such as Ethernet or Token Ring) addressed to the destination's physical address.

Indirect delivery occurs when the IP node (either the sending node or an IP router) forwards a packet to an intermediate node (an IP router) because the final destination is not on a directly attached network. The IP node encapsulates the IP datagram in a frame format, addressed to the IP router's physical address, for the Network Interface Layer (such as Ethernet or Token Ring).

IP routing is a combination of direct and indirect deliveries.

In the example in Figure 14, when sending packets to node B, node A will perform a direct delivery. When sending packets to node C, node A will perform an indirect delivery to Router 1. Router 1 will perform an indirect delivery to Router 2. Router 2 will perform a direct delivery to node C.

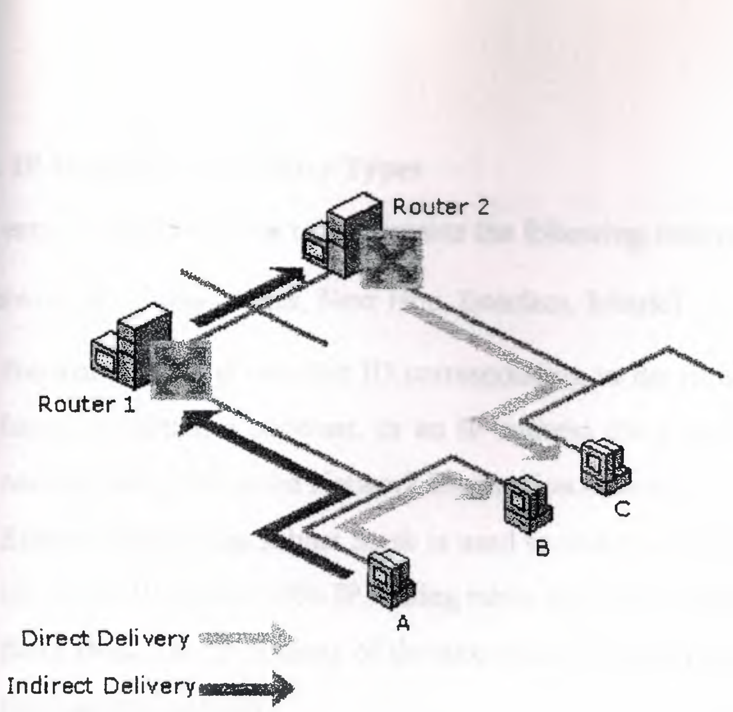


Figure 14 Direct and indirect deliveries

5.2. The IP Routing Table

A routing table is present on all IP nodes. The routing table stores information about IP networks and how they can be reached (either directly or indirectly). Since all IP nodes perform some form of IP routing, routing tables are not exclusive to IP routers. Any node loading the TCP/IP protocol will have a routing table. There are a series of default entries according to the configuration of the node and additional entries can be entered either manually through TCP/IP utilities or dynamically through interaction with routers.

When an IP packet is to be forwarded, the routing table is used to determine:

1. The forwarding IP address

   For a direct delivery, the forwarding IP address is the destination IP address in the IP packet. For an indirect delivery, the forwarding IP address is the IP address of a router.

2. The interface to be used for the forwarding

   The interface identifies the physical or logical interface such as a network interface card that will be used to forward the packet to either its destination or the next router.

5.3. IP Routing Table Entry Types

An entry in the IP routing table contains the following information:

[Network ID, Subnet Mask, Next Hop, Interface, Metric]

- **Network ID.** The network ID corresponding to the route. The Network ID can be class-based, a subnet, a supernet, or an IP address for a host route. In the Windows 2000 IP routing table, this is the Network Destination column.
- **Subnet Mask.** The subnet mask is used to match a destination IP address to the Network ID. In the Windows 2000 IP routing table, this is the Netmask column.
- **Next Hop.** The IP address of the next hop. In the Windows 2000 IP routing table, this is the Gateway column.
- **Interface.** An indication of which network interface will be used to forward the IP packet.
- **Metric.** A number used to indicate the cost of the route so the best route among possible multiple routes to the same destination can be selected. A common use of the metric is to indicate the number of hops (routers crossed) to the Network ID. The route with the lowest metric is the best route.

Routing table entries can be used to store the following types of routes:

- **Directly Attached Network IDs.** Routes for network IDs that are directly attached. For directly attached networks, the Next Hop field may be blank or contain the IP address of the interface on that network.
- **Remote Network IDs.** Routes for network IDs that are not directly attached but are available across other routers. For remote networks, the Next Hop field is the IP address of a local router in between the forwarding node and the remote network.
- **Host Routes.** A route to a specific IP address. Host routes allow routing to occur on a per-IP address basis. For host routes, the Network ID is the IP address of the specified host and the subnet mask is 255.255.255.255.
- **Default Route.** The default route is designed to be used when a more specific Network ID or host route is not found. The default route network ID is 0.0.0.0 with the subnet mask of 0.0.0.0.

5.4. The Route Determination Process

To determine which the routing table entry will be used for the forwarding decision, the following process is used:

- For each entry in routing table, perform a bit-wise logical AND between the Destination IP address and the Subnet Mask. Compare the result with the Network ID of the entry for a match.
- The list of matching routes is compiled. The route that has the longest match (the route that matched the most amount of bits with the destination IP address) is chosen. The longest matching route is the most specific route to the destination IP address. If multiple entries with the longest match are found (multiple routes to the same network ID, for example), the router uses the lowest metric to select the best route. If multiple entries exist that are the longest match and the lowest metric, the router is free to choose which routing table entry to use.

The result of the route determination process is the choice of a single route in the routing table. The route chosen yields a forwarding IP address (the next hop IP address) and an interface (the port). If the route determination process fails to find a route, IP declares a routing error. For the sending host, an IP routing error is internally indicated to the upper layer protocol such as TCP or UDP. For a router, an ICMP Destination Unreachable-Host Unreachable message is sent to the source host.

Example Routing Table for Windows 2000

Table 28 shows the default routing table for a Windows 2000 host (not a router). The host has a single network interface card and has the IP address 157.55.27.90, subnet mask 255.255.240.0 (/20), and default gateway of 157.55.16.1.

Table 28 **The Windows 2000 routing table**

Network Destination	Netmask	Gateway	Interface	Metric	Purpose
0.0.0.0	0.0.0.0	157.55.16.1	157.55.27.90	1	Default Route
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1	Loopback Network
157.55.16.0	255.255.240.0	157.55.27.90	157.55.27.90	1	Directly Attached Network
157.55.27.90	255.255.255.255	127.0.0.1	127.0.0.1	1	Local Host
157.55.255.255	255.255.255.255	157.55.27.90	157.55.27.90	1	Network Broadcast
224.0.0.0	224.0.0.0	157.55.27.90	157.55.27.90	1	Multicast Addresses
255.255.255.255	255.255.255.255	157.55.27.90	157.55.27.90	1	Limited Broadcast

- **Default Route.** The entry corresponding to the default gateway configuration is a Network Destination of 0.0.0.0 with a subnet mask of 0.0.0.0. Any destination IP address ANDed with 0.0.0.0 will result in 0.0.0.0. Therefore, for any IP address, the default route

   - If the TTL is 0, the router discards the packet and sends an ICMP Time Expired-TTL Expired message to the sender.

4. If the TTL is 1 or greater, IP updates the TTL field and calculates a new IP header checksum.
5. IP checks its routing table for the best route to the destination IP address in the IP datagram.
   - If no route is found, the router discards the packet and sends an ICMP Destination Unreachable-Host Unreachable message to the sender.
6. Based on the best route found, IP determines the forwarding IP address and the interface to be used for the forwarding.
7. IP hands the packet, the forwarding IP address, and the interface to ARP, and ARP forwards the packet to the appropriate MAC address.

This entire process is repeated at each router in the path between the source and destination host.

5.8. IP on the Destination Host

When a packet is received at the destination host, it is passed up to IP. IP on the destination host does the following:

1. IP verifies the IP header checksum.
   - If the IP header checksum fails, the IP packet is silently discarded.
2. IP verifies that the destination IP address in the IP datagram corresponds to an IP address assigned to the host.
   - If the destination IP address is not the host, the IP packet is silently discarded.
3. Based on the IP protocol field, IP passes the IP datagram without the IP header to the appropriate upper-level protocol.
   - If the protocol does not exist, ICMP sends a Destination Unreachable-Protocol Unreachable message back to the sender.
4. For TCP and UDP packets, the destination port is checked and TCP segment or UDP header is processed.
   - If no application exists for the UDP port number, ICMP sends a Destination Unreachable-Port Unreachable message back to the sender. If no application exists for the TCP port number, TCP sends a Connection Reset segment back to the sender.

5.9. Static and Dynamic IP Routers

In order for IP routing between routers to occur efficiently in the IP internetwork, routers must have explicit knowledge of remote network IDs or be properly configured with a default route. On large IP internetworks, one of the challenges faced by network administrators is how to maintain the routing tables on their IP routers so that IP traffic flow is traveling the best path and is fault-tolerant.

There are two ways of maintaining routing table entries on IP routers:

- **Manually**—Static IP routers have routing tables that do not change unless manually changed by a network administrator.
  Static routing relies on the manual administration of the routing table. Remote network IDs are not discovered by static routers and must be manually configured. Static routers are not fault tolerant. If a static router goes down, neighboring routers do not sense the fault and inform other routers.
- **Automatically**—Dynamic IP routers have routing tables that change automatically based on the communication of routing information with other routers. Dynamic routing employs the use of routing protocols, such as Routing Information Protocol (RIP) and Open Shortest Path First (OSPF), to dynamically update the routing table through the exchange of routing information between routers. Remote network IDs are discovered by dynamic routers and automatically entered into the routing table. Dynamic routers are fault-tolerant. If a dynamic router goes down, the fault is sensed by neighboring routers who propagate the changed routing information to the other routers in the internetwork.

For more information on routing principles, see the Unicast Routing Principles whitepaper.

CHAPTER SIX: PHYSICAL ADDRESS RESOLUTION

Based on the destination IP address and the route determination process, IP determines the forwarding IP address and interface to be used to forward the packet. IP then hands the IP packet, the forwarding IP address, and the interface to ARP.

If the forwarding IP address is the same as the destination IP address, then ARP performs a direct delivery. In a direct delivery, the MAC address corresponding to the destination IP address must be resolved.

If the forwarding IP address is not the same as the destination IP address, then ARP performs

Address of the interface sending the ARP Request, the IP address of the interface sending the ARP Request, and the forwarding IP address. ARP then broadcasts the ARP Request using the appropriate interface.

3. All hosts receive the broadcasted frame and the ARP Request is processed. If the receiving host's IP address matches the requested IP address (the forwarding IP address), its ARP cache is updated with the address mapping of the sender of the ARP Request.

   If the receiving host's IP address does not match the requested IP address, the ARP request is silently discarded.
4. An ARP Reply is formulated containing the requested MAC address and sent directly to the sender of the ARP Request.
5. When the ARP Reply is received by the sender of the ARP Request, it updates its ARP cache with the address mapping.

   Between the ARP Request and the ARP Reply, both hosts have each other's address mappings in their ARP caches.

The IP packet is sent to the MAC address of the forwarding host by addressing it to the resolved MAC address.

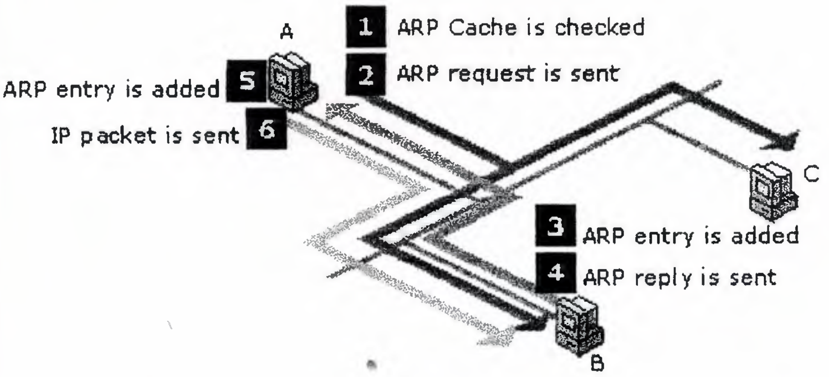


Figure 15 The ARP process

CONCLUSION

To ensure that computers in a network are able to communicate, they must share a common language known as a protocol. A protocol is a set of rules or standarts that enables communication between computers in a network. Transmission Control Protocol/Internet Protocol (TCP/IP) for Microsoft Windows 2000 offers a standard, routable enterprise networking protocol that is most complete and accepted protocol available TCP/IP is designed for communication across large-scale networks. The task involved in using TCP/IP in the communication process are distributed between protocols that are organized into four distinct layer of the TCP/IP stack. Each protocol in the TCP/IP stack has a distinct role in the communication process. This process involves multiple activities, including resolving user-friedly names to Internet Protocol (IP) addresses; determining the location of the destination computer. The Windows 2000 operating system has different storage locations where it keeps a record of user-friendly names mapped to their corresponding IP addresses. This mapping of the IP address of a computer can be stored in either a static or dynamic file, based on the type of name used.

Most network operating systems in use today offer TCP/IP support, and large networks rely on TCP/IP for much of their network traffic.

REFERENCES

I used Microsoft books.For the latest information on Windows 2000, check out our World Wide Web site at http://www.microsoft.com/ntserver/, the Windows NT Server Forum on MSNTM, and The Microsoft Network online service (GO WORD: MSNTS).

For more information on the Windows 2000 implementation of TCP/IP, please see the TCP/IP Implementation Details for Windows 2000 whitepaper. This whitepaper contains detailed information on Windows 2000 TCP/IP protocol components,