THE TCP/IP PROTOCOL SUITE

TCP/IP is an industry-standard suite of protocols designed for large internetworks spanning wide area network (WAN) links. TCP/IP was developed in 1969 by the U.S. Department of Defense Advanced Research Projects Agency (DARPA), the result of a 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.

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.

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.

TCP/IP Protocol Architecture

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.

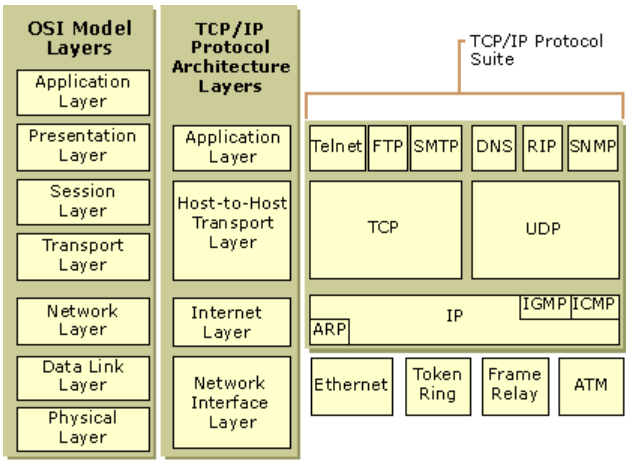

Figure 1 TCP/IP protocol architecture

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.

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.

Transport 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.

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.

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.

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.

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.

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

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 Request	Simple troubleshooting message used to check IP connectivity to a desired host.
Echo Reply	Response to an ICMP Echo Request.
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 Request messages and view statistics on the responses on a Windows NT-based computer, use the *ping* utility at a Windows NT 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
Network Unreachable	Sent by an IP router when a route to the destination network can not be found.
Host Unreachable	Sent by an IP router when a destination host on the destination network can not be found. This message is only used on connection-oriented network technologies (WAN links). IP routers on connectionless network technologies (such as Ethernet or Token Ring) do not send Host Unreachable messages.
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.

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 data was received by the other host. 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 integrity of the TCP header and the TCP data.

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

For a complete list of assigned TCP ports, see RFC 1700.

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. 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.

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 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.

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)

For a complete list of assigned UDP ports, see RFC 1700.

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.

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).

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. For Windows NT–based computers, the host name does not have to match the Windows NT computer name.

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.

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.

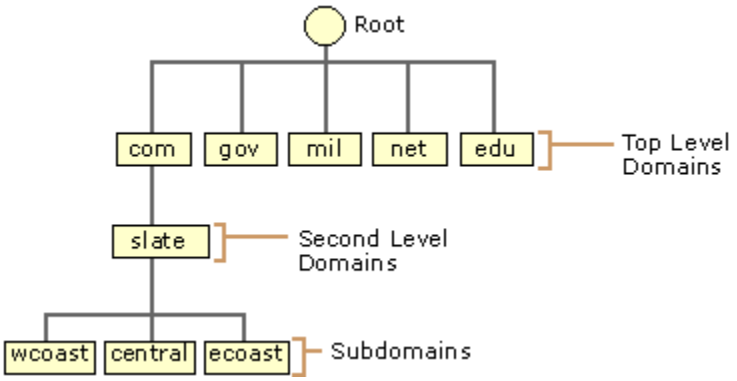

Figure 11 The Domain Name System

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.slate.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.
- **slate** is the second-level domain, indicating the Slate magazine company.
- **wcoast** is a subdomain of slate.com indicating the West Coast division of the Slate magazine company.
- **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.