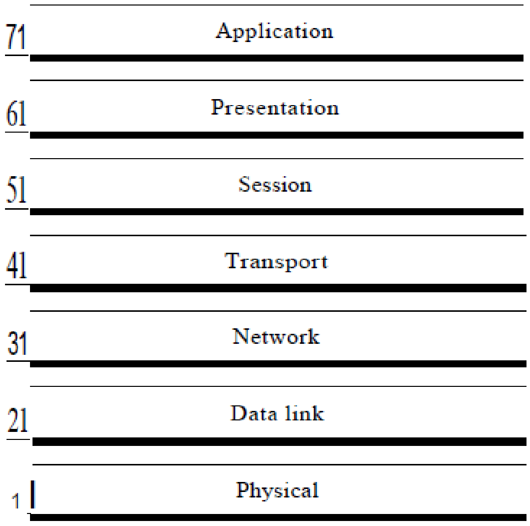
UNIT-2

The OSI Model

Established in 1947, the International Standards Organization (ISO) is a multinational body dedicated to worldwide agreement on international standards. An ISO standard that covers all aspects of network communications is the Open Systems Interconnection model. It was first introduced in the late 1970s. An open system is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture.

The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network.



Seven layers of the OSI model

Layers in the OSI Model

Physical Layer

The physical layer coordinates the functions required to carry a bit stream over a physical medium. It deals with the mechanical and electrical specifications of the interface and transmission medium. It also defines the procedures and functions that physical devices and interfaces have to perform for transmission to Occur.

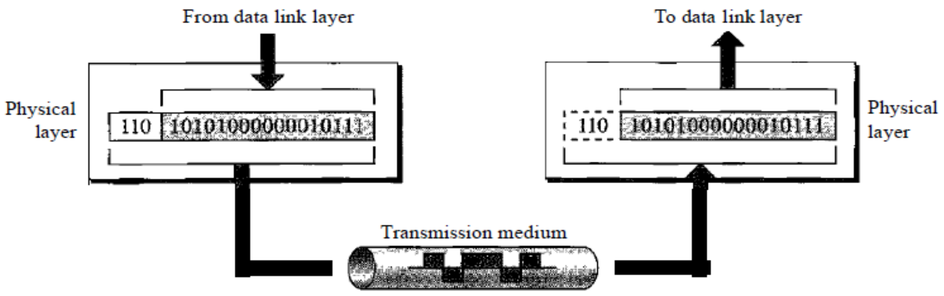


Fig.Physical layer

The physical layer is also concerned with the following:

- **Physical characteristics of interfaces and medium.** The physical layer defines the characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium.

- **Representation of bits.** The physical layer data consists of a stream of bits (sequence of Os or 1s) with no interpretation. To be transmitted, bits must be encoded into signals--electrical or optical. The physical layer defines the type of encoding (how Os and I s are changed to signals).

- **Data rate.** The transmission rate-the number of bits sent each second-is also defined by the physical layer. In other words, the physical layer defines the duration of a bit, which is how long it lasts.

- **Synchronization of bits.** The sender and receiver not only must use the same bit rate but also must be synchronized at the bit level. In other words, the sender and the receiver clocks must be synchronized.

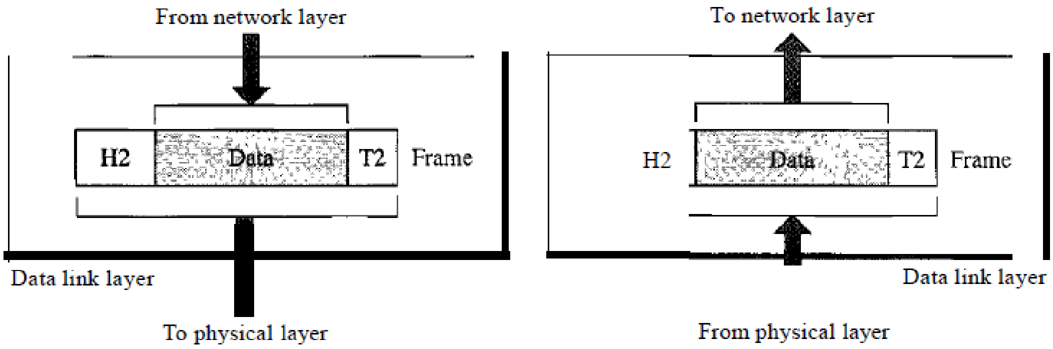
- **Line configuration.** The physical layer is concerned with the connection of devices to the media. In a point-to-point configuration, two devices are connected through a dedicated link. In a multipoint configuration, a link is shared among several devices.

- **Physical topology.** The physical topology defines how devices are connected to make a network. Devices can be connected by using a mesh topology (every device is connected to every other device), a star topology (devices are connected through a central device), a ring topology (each device is connected to the next, forming a ring), a bus topology (every device is on a common link), or a hybrid topology (this is a combination of two or more topologies).

- **Transmission mode.** The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex. In simplex mode, only one device can send; the other can only receive. The simplex mode is a one-way communication. In the half-duplex mode, two devices can send and receive, but not at the same time. In a full-duplex (or simply duplex) mode, two devices can send and receive at the same time.

Data Link Layer

The data link layer transforms the physical layer, a raw transmission facility, to a reliable link. It makes the physical layer appear error-free to the upper layer (network layer). Figure shows the relationship of the data link layer to the network and physical layers.



Data Link Layer.

The data link layer is responsible for moving frames from one hop (node) to the next.
Other responsibilities of the data link layer include the following:

- **Framing.** The data link layer divides the stream of bits received from the network layer into manageable data units called frames.

- **Physical addressing.** If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame. If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects the network to the next one.

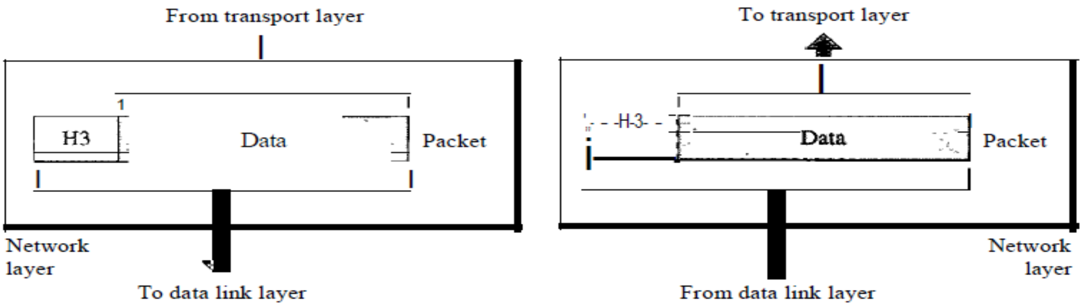
- **Flow control.** If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

- **Error control.** The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.

- **Access control.** When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Network Layer

The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links). The network layer ensures that each packet gets from its point of origin to its final destination. If two systems are connected to the same link, there is usually no need for a network layer. However, if the two systems are attached to different networks (links) with connecting devices between the networks (links), there is often a need for the network layer to accomplish source-to-destination delivery.



Network Layer.

Responsibilities of the network layer include the following:

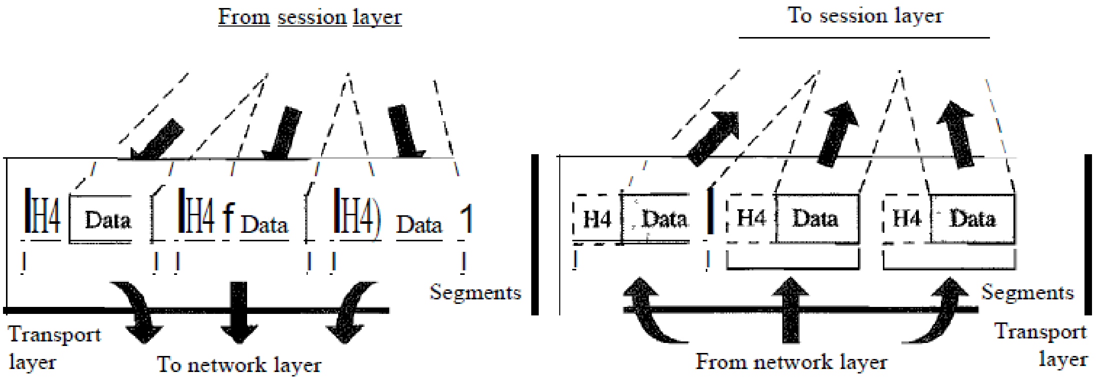
- **Logical addressing.** The physical addressing implemented by the data link layer handles the addressing problem locally. If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems. The network layer adds a header to the packet coming from the upper layer that, among other things, includes the logical addresses of the sender and receiver.

- **Routing.** When independent networks or links are connected to create *internetworks* (network of networks) or a large network, the connecting devices (called *routers* or *switches)* route or switch the packets to their final destination. One of the functions of the network layer is to provide this mechanism.

Transport Layer

The transport layer is responsible for process-to-process delivery of the entire message. A process is an application program running on a host. Whereas the network layer oversees source-to-destination delivery of individual packets, it does not recognize any relationship between those packets. It treats each one independently, as though each piece belonged to a separate message, whether or not it does. The transport layer, on the other hand, ensures that the whole message arrives intact and in order, overseeing both error control and flow control at the source-to-destination level.

.



Transport Layer.

Responsibilities of the transport layer include the following:

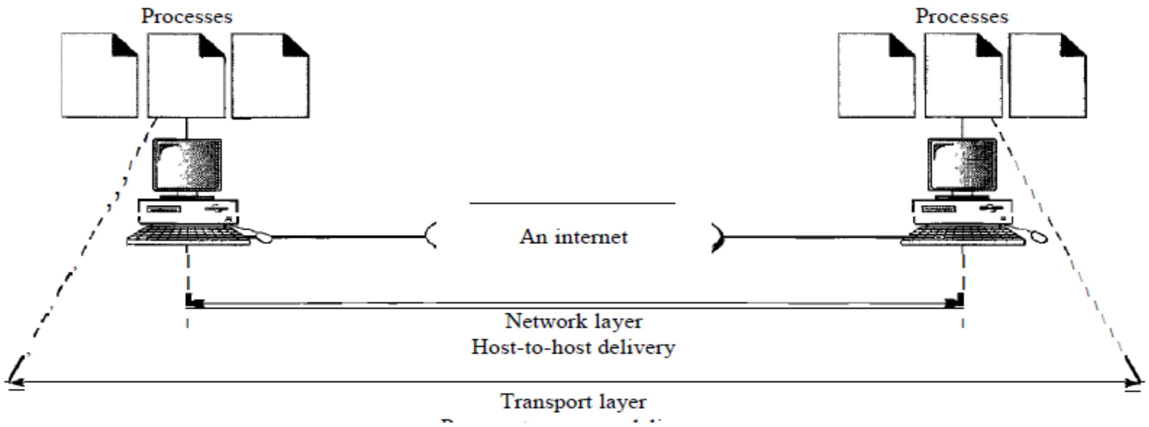
- **Service-point addressing:** Computers often run several programs at the same time. For this reason, source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to a specific process (running program) on the other. The transport layer header must therefore include a type of address called a *service-point address* (or port address). The network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct process on that computer.

- **Segmentation and reassembly:** A message is divided into transmittable segments, with each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in transmission.

- **Connection control:** The transport layer can be either connectionless or connection oriented. A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred, the connection is terminated.

- **Flow control:** Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.

- **Error control:** Like the data link layer, the transport layer is responsible for error control. However, error control at this layer is performed process-to process rather than across a single link. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication). Error correction is usually achieved through retransmission. Illustration of process-to-process delivery by the transport layer.



Reliable process-to-process delivery of a message.

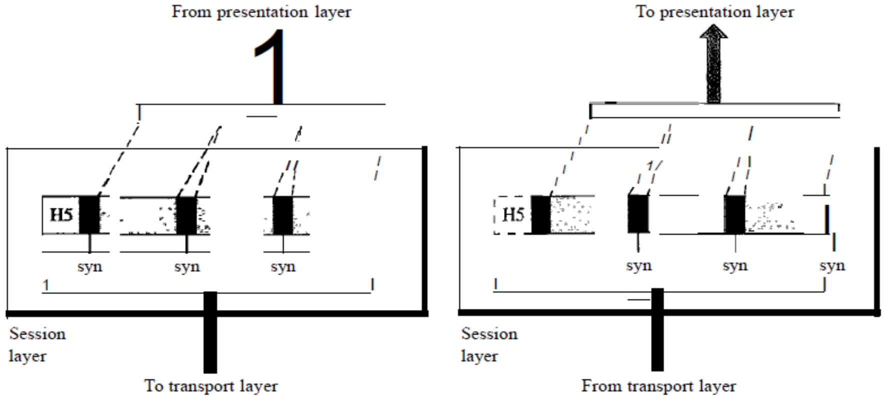
Session Layer

The services provided by the first three layers (physical, data link, and network) are not sufficient for some processes. The session layer is the network *dialog controller*. It establishes, maintains, and synchronizes the interaction among communicating systems. The session layer is responsible for dialog control and synchronization.

Specific responsibilities of the session layer include the following:

- **Dialog control:** The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either half duplex (one way at a time) or full-duplex (two ways at a time) mode.

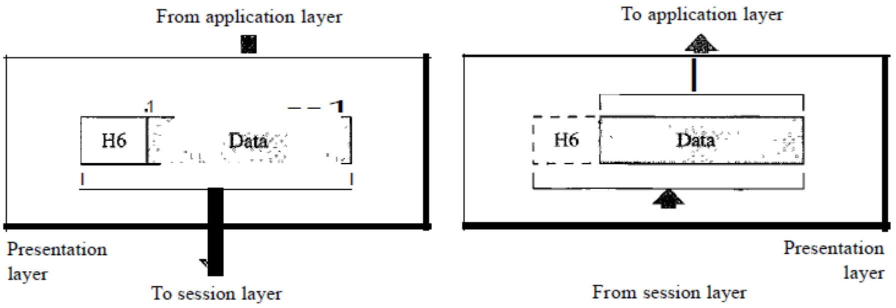
- **Synchronization:** The session layer allows a process to add checkpoints, or synchronization points, to a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, the only pages that need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need not be resent. Figure illustrates the relationship of the session layer to the transport and presentation layers.



Session layer

Presentation Layer

The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems. Figure shows the relationship between the presentation layer and the application and session layers.



Presentation layer

Specific responsibilities of the presentation layer include the following:

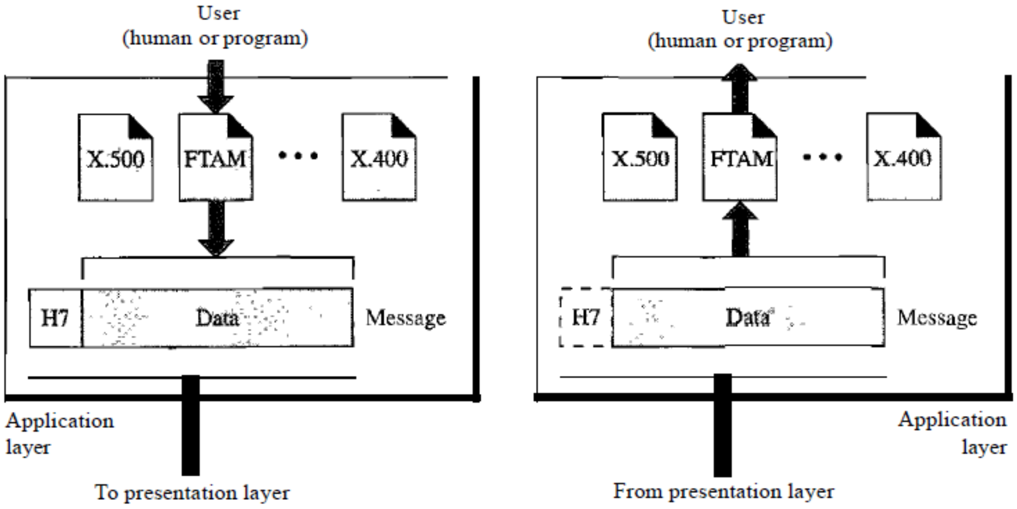
- **Translation:** The processes (running programs) in two systems are usually exchanging information in the form of character strings, numbers, and so on. The information must be changed to bit streams before being transmitted. Because different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information from its sender-dependent format into a common format. The presentation layer at the receiving machine changes the common format into its receiver-dependent format.

- **Encryption:** To carry sensitive information, a system must be able to ensure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.

- **Compression:** Data compression reduces the number of bits contained in the information. Data compression becomes particularly important in the transmission of multimedia such as text, audio, and video.

Application Layer

The application layer enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as electronic mail, remote file access and transfer, shared database management, and other types of distributed information services.



Application layer

Specific services provided by the application layer include the following:

- **Network virtual terminal:** A network virtual terminal is a software version of a physical terminal, and it allows a user to log on to a remote host. To do so, the application creates a software emulation of a terminal at the remote host. The user's computer talks to the software terminal which, in turn, talks to the host, and vice versa. The remote host believes it is communicating with one of its own terminals and allows the user to log on.

- **File transfer, access, and management:** This application allows a user to access files in a remote host (to make changes or read data), to retrieve files from a remote computer for use in the local computer, and to manage or control files in a remote computer locally.

- **Mail services:** This application provides the basis for e-mail forwarding and storage.

- **Directory services:** This application provides distributed database sources and access for global information about various objects and services.

Internetworking devices

Various devices are used to connect network of a computer The most common devices are:

1) **Bridges 2) Routers 3) Repeaters 4) Hub 5) Gateways**

   BRIDGES

- Operates in both the physical and data link layers
- As physical layer device, it regenerates the signal it receives.
- As a data link layer device, the bridge can check the MAC (source and destination) address contained in the frame.
- A bridge has a table used in filtering decisions.
- Difference in functionality between a bridge and a repeater?
  - A bridge has filtering capability.
  - It can check the destination address of a frame and decide if the frame should be forwarded or dropped.
  - If the frame is to be forwarded, the decision must specify the port.
  - A bridge has a table that maps addresses to ports.
- A bridge does not change the physical (MAC) addresses in a frame.

Router

Routers are devices which connect two are more networks that use similar protocol. A router consists of hard ware and software. Hard ware can be a computer is specific device. Software consists of special management program that controls flow of data between networks. Routers operate at a network layer of O.S.I model.

Routers use logical and physical address to connect two or more logically separate network. They make this connection by organizing the large network into logical network segment (some times small sub network or sub nets). Each of these sub nets is given a logical address. Data is grouped into packets or block of data. Each packet in addition to having a physical device address, has a logical address. The network address allows routers to calculate more accurately and efficiently the path of the computer.

Repeaters

Repeaters are used within network to extend the length of communication. Data process through transmission media in the farm of waves or signals. The transmission media weaken signals that move through it. The weakening of signal is called **attenuation.** If the data is to be transmitted beyond the maximum length of a communication media, signals have amplified. The devices that are used to amplify the signals are called repeaters. Repeaters work at the physical layer of OSI model. Repeaters are normally two ports boxes that connect two segments. As a signal comes in one port, it is regenerated and send out to the other port.

The signal is read as 1s and 0s. As 1s and 0s are transmitted, the noise can be cleaned out.

Hub

Hubs are basically multi ports repeaters for U.T.P cables. Some hubs have ports for other type of cable such as coaxial cable. Hubs range in size from four ports up to and for specific to the network types. These are some hubs which are

I. Passive Hub

II. Active Hub

III. Switch/ Intelligent Hub

Passive Hub: It provides no signal regeneration. They are simply cables connected together so that the signal is broken out to other nodes without regeneration. These are not used often today because of loss of cable length that is allowed.

Active Hub: It acts as repeaters and regenerates the data signals to all ports. They have no real intelligence to tell whether the signal needs to go to all ports that is blindly repeated.

Switch Hub: Switches are multi ports bridges. They filter traffic between the ports on the switch by using the address of computers transmitting to them.

Switches can be used when data performance is needed or when collision need to be reduce.

Gateways

Gateways are devices which connect two are more networks that use different protocols. They are similar in function to routes but they are more powerful and intelligent devices. A gateway can actually convert data so that network with an application on a computers on the other side of the gateway e.g a get way can receive email messages in one format in convert them into another format. Gateway can operate at all seven layer of OSI model. Since Gateway perform data conversion so they are slower in speed and very expensive devices.

IP address sing

Static vs. dynamic IP addressing

An Internet Protocol (IP) address is a unique number assigned to every device on a network. Just as a street address determines where a letter should be delivered, an IP address identifies computers on the Internet. Network devices use IP addresses to communicate with each other.

Although computers use IP addresses - numbers - to communicate, it's easier for people to remember words than numbers. The Internet uses DNS (Domain Name System) to enable people to use words instead of numbers for Internet addresses. You can think of DNS as an address book for the Internet. DNS maps domain names to IP addresses.

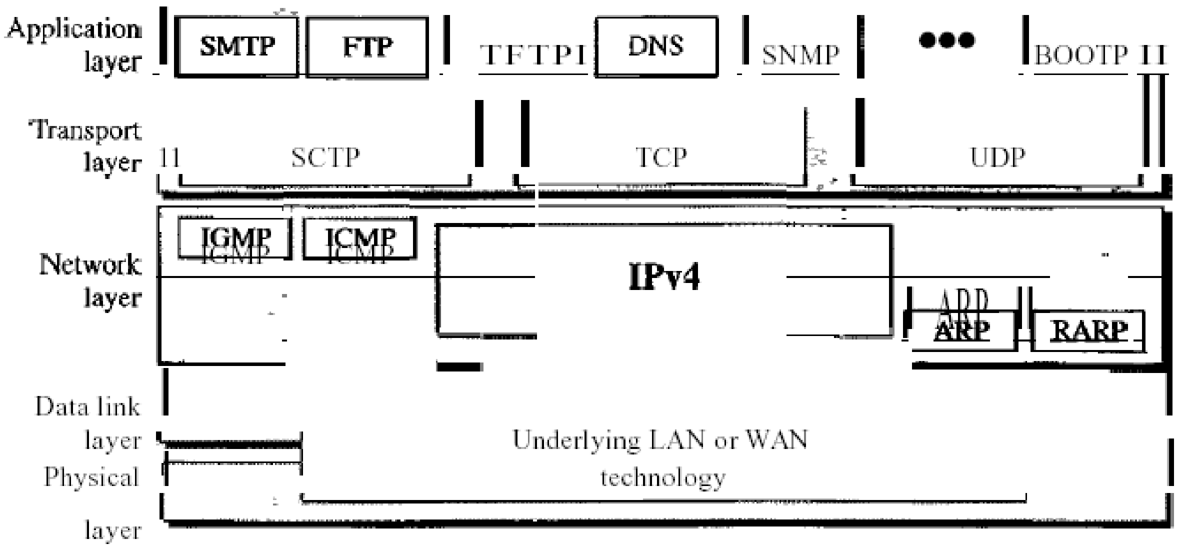
Difference between static and dynamic

IPs When a device is assigned a static IP address, it does not change. The device always has the same IP address. Most devices use dynamic IP addresses, which are assigned by the network when they connect. These IP addresses are temporary, and can change over time.

.

IP (Internet Protocols) - IPv4, IPv6

The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols. Figure shows the position of IPv4 in the suite.

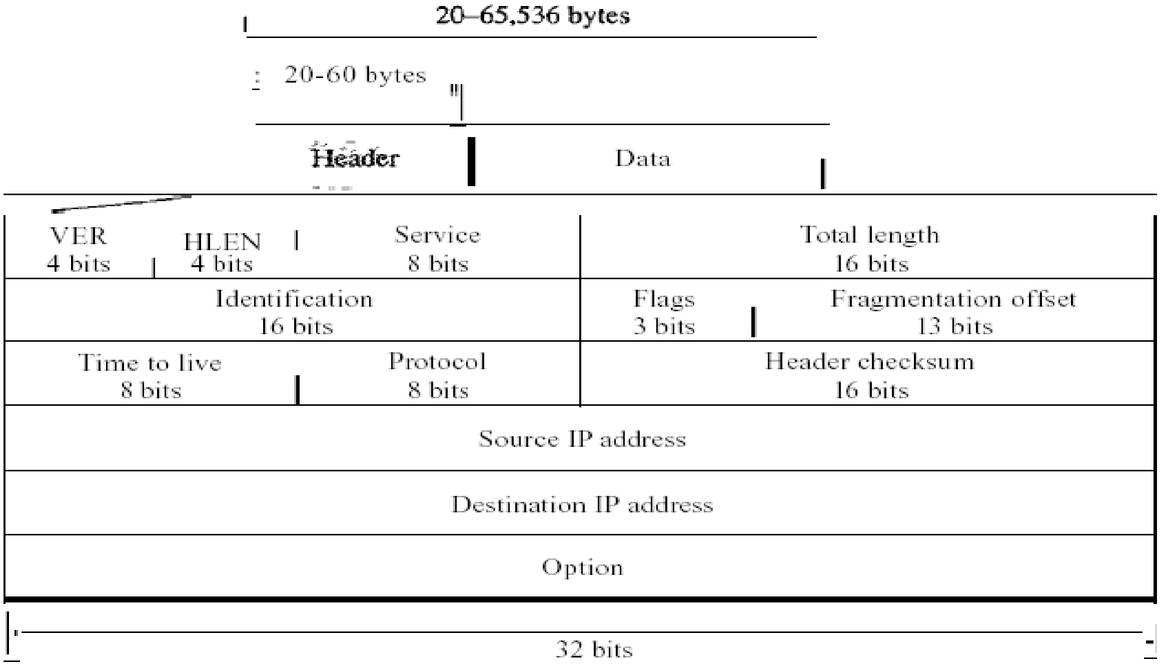


IPv4 is an unreliable and connectionless datagram protocol-a best-effort delivery service. The term best-effort means that IPv4 provides no error control or flow control (except for error detection on the header). IPv4 assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees.

If reliability is important, IPv4 must be paired with a reliable protocol such as TCP. An example of a more commonly understood best-effort delivery service is the post office. The post office does its best to deliver the mail but does not always succeed. If an unregistered letter is lost, it is up to the sender or would-be recipient to discover the loss and rectify the problem. The post office itself does not keep track of every letter and cannot notify a sender of loss or damage.

Datagram

Packets in the IPv4 layer are called datagrams. Figure shows the IPv4 datagram format.



Datagram is a variable-length packet consisting of two parts: header and data. The header is 20 to 60 bytes in length and contains information essential to routing delivery. It is customary in TCP/IP to show the header in 4-byte sections. A brief description of each field is in order.

Version (VER) - This 4-bit field defines the version of the IPv4 protocol. Currently the version is 4. This field tells the IPv4 software running in the processing machine that the datagram has the format of version 4.

Header length (HLEN) - This 4-bit field defines the total length of the datagram header in 4-byte words. This field is needed because the length of the header is variable (between 20 and 60 bytes). When there are no options, the header length is 20 bytes, and the value of this field is 5 (5 x 4 = 20). When the option field is at its maximum size, the value of this field is 15 (15 x 4 = 60).

Services - This field, previously called service type, is now called differentiated services. In Service Type Interpretation, the first 3 bits are called precedence bits. The next 4 bits are called type of service (TOS) bits, and the last bit is not used.

In the Differentiated Services this interpretation, the first 6 bits make up the code point subfield, and the last 2 bits are not used. The code point subfield can be used in two different ways.

Total length - This is a In-bit field that defines the total length (header plus data) of the IPv4 datagram in bytes. To find the length of the data coming from the upper layer, subtract the header length from the total length. The header length can be found by multiplying the value in the HLEN field by 4.

Length of data =total length - header length

Identification - This field is used in fragmentation.

Flags - This field is used in fragmentation.

Fragmentation offset - This field is used in fragmentation

Time to live - A datagram has a limited lifetime in its travel through an internet. This field was originally designed to hold a timestamp, which was decremented by each visited router. The datagram was discarded when the value became zero.

Protocol - This 8-bit field defines the higher-level protocol that uses the services of the IPv4 layer. An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP. This field specifies the final destination protocol to which the IPv4 datagram is delivered.

Checksum. The checksum concept and its calculation are used to detect errors

Source address. This 32-bit field defines the IPv4 address of the source. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.

Destination address. This 32-bit field defines the IPv4 address of the destination. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.

IPv6 Protocol

The network layer protocol in the TCP/IP protocol suite is currently IPv4 (Internetworking Protocol, version 4). IPv4 provides the host-to-host communication between systems in the Internet.

IPv4 has some deficiencies that make it unsuitable for the fast-growing Internet which are listed below..

- Address depletion is still a long-term problem in the Internet.
- The Internet must accommodate real-time audio and video transmission. This type of transmission requires minimum delay strategies and reservation of resources not provided in the IPv4 design.
- No encryption or authentication is provided by IPv4.

To overcome these deficiencies, IPv6 (Internetworking Protocol, version 6), also known as IPng (Internetworking Protocol, next generation), was proposed and is now a standard. In IPv6, the Internet protocol was extensively modified to accommodate the unforeseen growth of the Internet. The format and the length of the IP address were changed along with the packet format. Related protocols, such as ICMP, were also modified. Other protocols in the network layer, such as ARP, RARP, and IGMP, were either deleted or included in the ICMPv6 protocol.

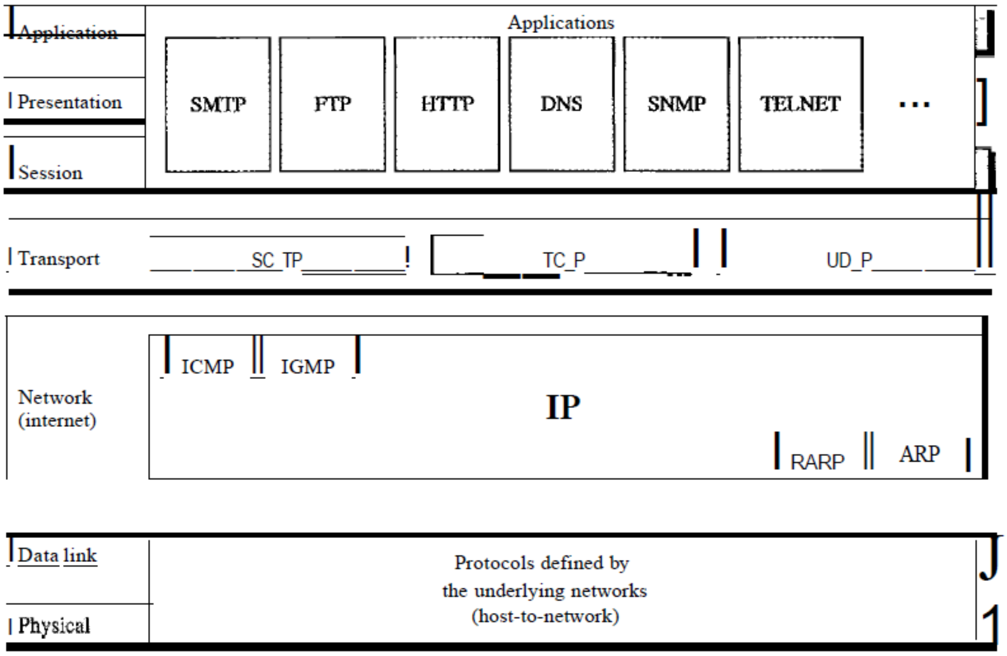
Advantages

The next-generation IP, or IPv6, has some advantages over IPv4 that can be summarized as follows:

- Larger address space - An IPv6 address is 128 bits long, Compared with the 32-bit address of IPv4, this is a huge (296) increase in the address space.
- Better header format - IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the upper-layer data.
- New options - IPv6 has new options to allow for additional functionalities.
- Allowance for extension - IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
- Support for resource allocation - In IPv6, the type-of-service field has been removed, but a mechanism (called flow label) has been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.
- Support for more security. The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet

TCP/IP Protocol Suite

The TCP/IP protocol suite was developed prior to the OSI model. Therefore, the layers in the TCP/IP protocol suite do not exactly match those in the OSI model. The TCP/IP protocol suite is made of five layers: physical, data link, network, transport, and application. The first four layers provide physical standards, network interfaces, internetworking, and transport functions that correspond to the first four layers of the OSI model. The three top most layers in the OSI model, however, are represented in TCP/IP by a single layer called the application layer.



TCP/IP and OSI model

TCP/IP is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality; however, the modules are not necessarily interdependent. Whereas the OSI model specifies which functions belong to each of its layers, the layers of the *TCP/IP* protocol suite contain relatively independent protocols that can be mixed and matched depending on the needs of the system. The term *hierarchical* means that each upper-level protocol is supported by one or more lower-level protocols.

At the transport layer, *TCP/IP* defines three protocols: Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Stream Control Transmission Protocol (SCTP). At the network layer, the main protocol defined by TCP/IP is the Internetworking Protocol (IP); there are also some other protocols that support data movement in this layer.

Physical and Data Link Layers

At the physical and data link layers, *TCPIIP* does not define any specific protocol. It supports all the standard and proprietary protocols. A network in a *TCPIIP* internetwork can be a local-area network or a wide-area network.

Network Layer (Internet layer)

At the network layer (or, more accurately, the internetwork layer), *TCP/IP* supports the Internetworking Protocol. IP, in turn, uses four supporting protocols: ARP, RARP, ICMP, and IGMP.

Internetworking Protocol (IP)

The Internetworking Protocol (IP) is the transmission mechanism used by the TCP/IP protocols. It is an unreliable and connectionless protocol-a best-effort delivery service. The term *best effort* means that IP provides no error checking or tracking. IP assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees. IP transports data in packets called datagrams, each of which is transported separately. Datagrams can travel along different routes and can arrive out of sequence or be duplicated. IP does not keep track of the routes and has no facility for reordering datagrams once they arrive at their destination. IP provides bare-bones transmission functions that free the user to add only those facilities necessary for a given application and thereby allows for maximum efficiency.

Address Resolution Protocol

The Address Resolution Protocol (ARP) is used to associate a logical address with a physical address. On a typical physical network, such as a LAN, each device on a link is identified by a physical or station address, usually imprinted on the network interface card (NIC). ARP is used to find the physical address of the node when its Internet address is known.

Reverse Address Resolution Protocol

The Reverse Address Resolution Protocol (RARP) allows a host to discover its Internet address when it knows only its physical address. It is used when a computer is connected to a network for the first time or when a diskless computer is booted.

Internet Control Message Protocol

The Internet Control Message Protocol (ICMP) is a mechanism used by hosts and gateways to send notification of datagram problems back to the sender. ICMP sends query and error reporting messages.

Internet Group Message Protocol

The Internet Group Message Protocol (IGMP) is used to facilitate the simultaneous transmission of a message to a group of recipients.

Transport Layer

Traditionally the transport layer was represented in *TCP/IP* by two protocols: TCP and UDP. IP is a host-to-host protocol, meaning that it can deliver a packet from one physical device to another. UDP and TCP are transport level protocols responsible for delivery of a message from a process (running program) to another process. A new transport layer protocol, SCTP, has been devised to meet the needs of some newer applications.

Transmission Control Protocol

The Transmission Control Protocol (TCP) provides full transport-layer services to applications. TCP is a reliable stream transport protocol. The term *stream,* in this context, means connection-oriented: A connection must be established between both ends of a transmission before either can transmit data.

At the sending end of each transmission, TCP divides a stream of data into smaller units called *segments.* Each segment includes a sequence number for reordering after receipt, together with an acknowledgment number for the segments received. Segments are carried across the internet inside of IP datagrams. At the receiving end, TCP collects each datagram as it comes in and reorders the transmission based on sequence numbers.

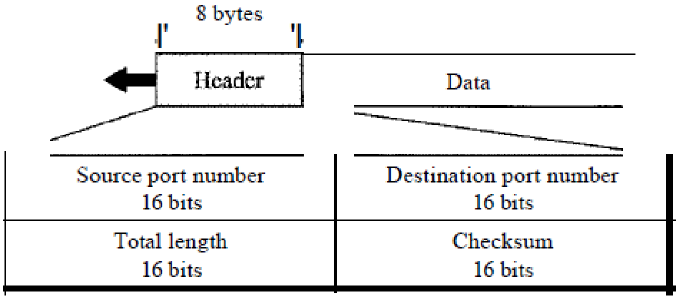
Stream Control Transmission Protocol

The Stream Control Transmission Protocol (SCTP) provides support for newer applications such as voice over the Internet. It is a transport layer protocol that combines the best features of UDP and TCP.

User Datagram Protocol (UDP)

The User Datagram Protocol (UDP) is the simpler of the two standard TCPIIP transport protocols. It is a process-to-process protocol that adds only port addresses, checksum error control, and length information to the data from the upper layer.

The User Datagram Protocol (UDP) is called a connectionless, unreliable transport protocol. It does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication. Also, it performs very limited error checking.
If a process wants to send a small message and does not care much about reliability, it can use UDP. Sending a small message by using UDP takes much less interaction between the sender and receiver than using TCP or SCTP.



Source port number 16 bits	Destination port number 16 bits
Total length 16 bits	Checksum 16 bits

User datagram format.

Application Layer - Protocols

The main task of the Internet is to provide services for users. Among the most popular applications is remote logging, electronic mail, and file transfer.

Remote Logging

In the Internet, users may want to run application programs at a remote site and create results that can be transferred to their local site. For example, students may want to connect to their university computer lab from their home to access application programs for doing homework assignments or projects. One way to satisfy that demand and others is to create a client/server application program for each desired service. Programs such as file transfer programs (FTPs), e-mail (SMTP), and so on are currently available. However, it would be impossible to write a specific client/server program for each demand. The better solution is a general-purp<Dse client/server program that lets a user access any application program on a remote computer; in other words, allow the user to log on to a remote computer. After logging on, a user can use the services available on the remote computer and transfer the results back to the local computer.

Telnet

TELNET is an abbreviation for ***TErminaL NETwork***. It is the standard TCP/IP protocol for virtual terminal service as proposed by the International Organization for Standards (ISO). TELNET enables the establishment of a connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system.

TELNET is a general-purpose client/server application program.

File Transfer

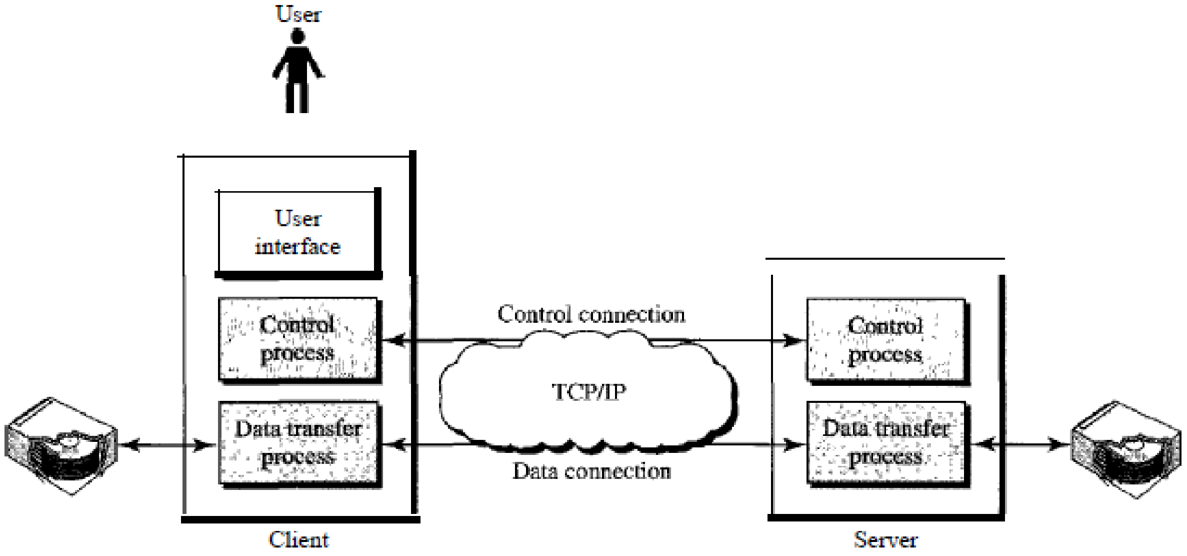
Transferring files from one computer to another is one of the most common tasks expected from a networking or internetworking environment. As a matter of fact, the greatest volume of data exchange in the Internet today is due to file transfer.

File Transfer Protocol (FTP)

File Transfer Protocol (FTP) is the standard mechanism provided by *TCP/IP* for copying a file from one host to another. Although transferring files from one system to another seems simple and straightforward, some problems must be dealt with first. For example, two systems may use different file name conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures. All these problems have been solved by FTP in a very simple and elegant approach.

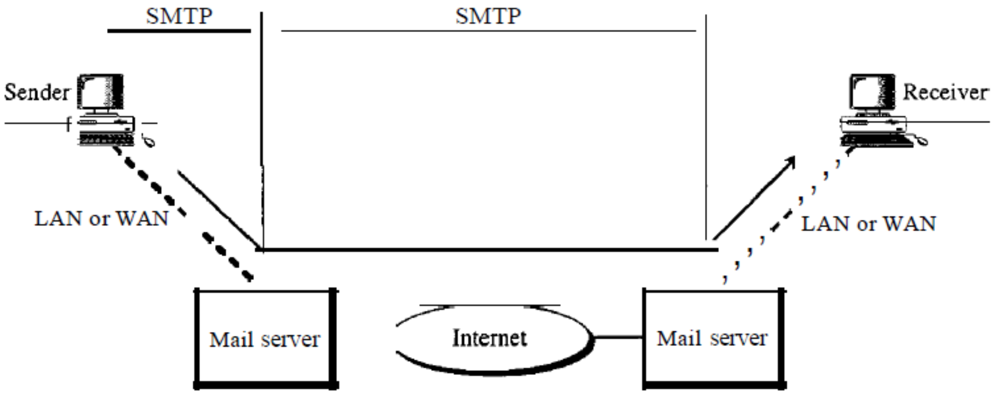
FTP uses two well-known TCP ports: Port 21 is used for the control connection, and port 20 is used for the data connection.

Figure shows the basic model of FTP. The client has three components: user interface, client control process, and the client data transfer process. The server has two components: the server control process and the server data transfer process. The control connection is made between the control processes. The data connection is made between the data transfer processes.



Message Transfer Agent: SMTP

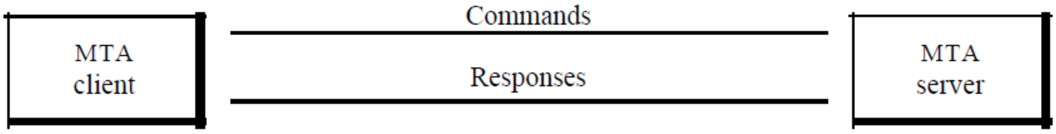
SMTP: The actual mail transfer is done through message transfer agents. To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA. The formal protocol that defines the MTA client and server in the Internet is called the Simple Mail Transfer Protocol (SMTP). As we said before, two pairs of MTA client/server programs are used in the most common situation (fourth scenario). Figure 26.16 shows the range of the SMTP protocol in this scenario.



The control connection remains connected during the entire interactive FTP session. The data connection is opened and then closed for each file transferred. It opens each time commands that involve transferring files are used, and it closes when the file is transferred. In other words, when a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.

Commands and Responses

SMTP uses commands and responses to transfer messages between an MTA client and
an MTA server (see Figure 26.17).



Message Access Agent: **POP** and IMAP

The first and the second stages of mail delivery use SMTP. However, SMTP is not involved in the third stage because SMTP is a *push* protocol; it pushes the message from the client to the server. In other words, the direction of the bulk: data (messages) is from the client to the server. On the other hand, the third stage needs a *pull* protocol; the client must pull messages from the server. The direction of the bulk data is from the server to the client. The third stage uses a message access agent.

Currently two message access protocols are available: Post Office Protocol, version 3 (POP3) and Internet Mail Access Protocol, version 4 (IMAP4). Figure 26.19 shows the position of these two protocols in the most common situation (fourth scenario).

POP3

Post Office Protocol, version 3 (POP3) is simple and limited in functionality. The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.

Mail access starts with the client when the user needs to download e-mail from the mailbox on the mail server. The client opens a connection to the server on TCP port 110. It then sends its user name and password to access the mailbox. The user can then list and retrieve the mail messages, one by one. Figure 26.20 shows an example of downloading using POP3.

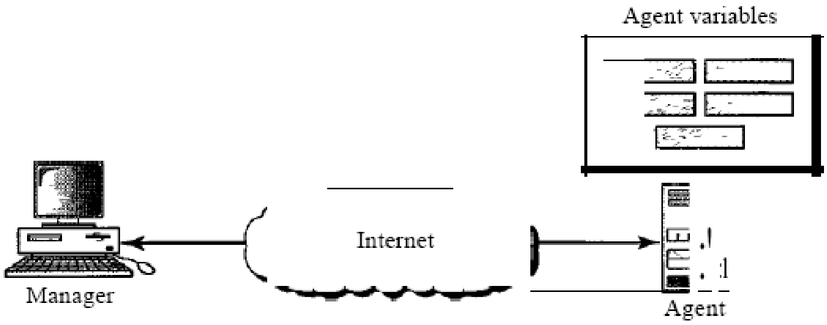
POP3 has two modes: the delete mode and the keep mode. In the delete mode, the mail is deleted from the mailbox after each retrieval. In the keep mode, the mail remains in the mailbox after retrieval. The delete mode is normally used when the user is working at her permanent computer and can save and organize the received mail after reading or replying. The keep mode is normally used when the user accesses her mail away from her primary computer (e.g., a laptop). The mail is read but kept in the system for later retrieval and organizing.

Simple Network Management Protocol (SNMP)

The Simple Network Management Protocol (SNMP) is a framework for managing devices in an internet using the TCPIIP protocol suite. It provides a set of fundamental operations for monitoring and maintaining an internet.

Concept

SNMP uses the concept of manager and agent. That is, a manager, usually a host, controls and monitors a set of agents, usually routers (see Figure).



SNMP concept.

SNMP is an application-level protocol in which a few manager stations control a set of agents. The protocol is designed at the application level so that it can monitor devices made by different manufacturers and installed on different physical networks.

Managers and Agents

A management station, called a manager, is a host that runs the SNMP client program. A managed station, called an agent, is a router (or a host) that runs the SNMP server program. Management is achieved through simple interaction between a manager and an agent. The agent keeps performance information in a database. The manager has access to the values in the database. For example, a router can store in appropriate variables the number of packets received and forwarded. The manager can fetch and compare the values of these two variables to see if the router is congested or not.

The manager can also make the router perform certain actions. For example, a router periodically checks the value of a reboot counter to see when it should reboot itself. It reboots itself, for example, if the value of the counter is O. The manager can use this feature to reboot the agent remotely at any time. It simply sends a packet to force a 0 value in the counter.

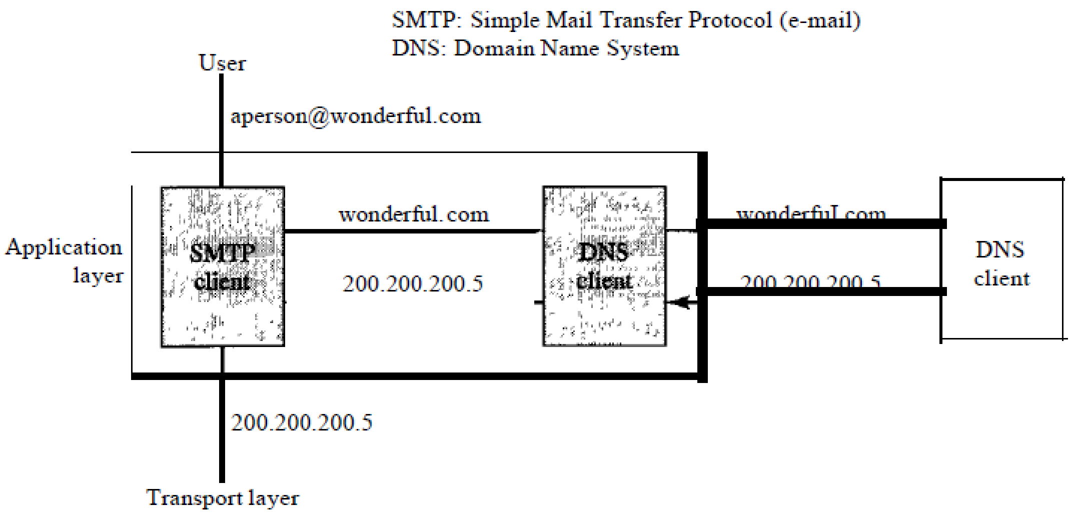
Agents can also contribute to the management process. The server program running on the agent can check the environment, and if it notices something unusual, it can send a warning message, called a trap, to the manager.

HTTP

The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web. HTTP functions as a combination of FTP and SMTP. It is similar to FfP because it transfers files and uses the services of TCP. However, it is much simpler than FfP because it uses only one TCP connection. There is no separate control connection; only data are transferred between the client and the server. HTTP is like SMTP because the data transferred between the client and the server look like SMTP messages.

Domain Name System (DNS)

The Domain Name System (DNS) is a supporting program that is used by other programs such as e-mail. Figure shows an example of how a DNS client/server program can support an e-mail program to find the IP address of an e-mail recipient. A user of an e-mail program may know the e-mail address of the recipient; however, the IP protocol needs the IP address. The DNS client program sends a request to a DNS server to map the e-mail address to the corresponding IP address.

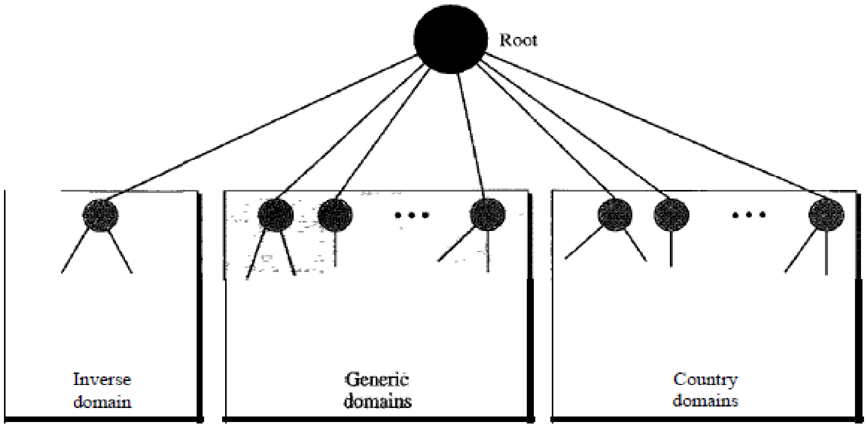


Example of using the DNS service

When the Internet was small, mapping was done by using a host file. The host file had only two columns: name and address. Every host could store the host file on its disk and update it periodically from a master host file. When a program or a user wanted to map a name to an address, the host consulted the host file and found the mapping. But we know that this would create a huge amount of traffic on the Internet. Another solution, the one used today, is to divide this huge amount of information into smaller parts and store each part on a different computer. In this method, the host that needs mapping can contact the closest computer holding the needed information. This method is used by the Domain Name System (DNS).

DNS in the Internet

DNS is a protocol that can be used in different platforms. In the Internet, the domain name space (tree) is divided into three different sections: generic domains, country domains, and the inverse domain.

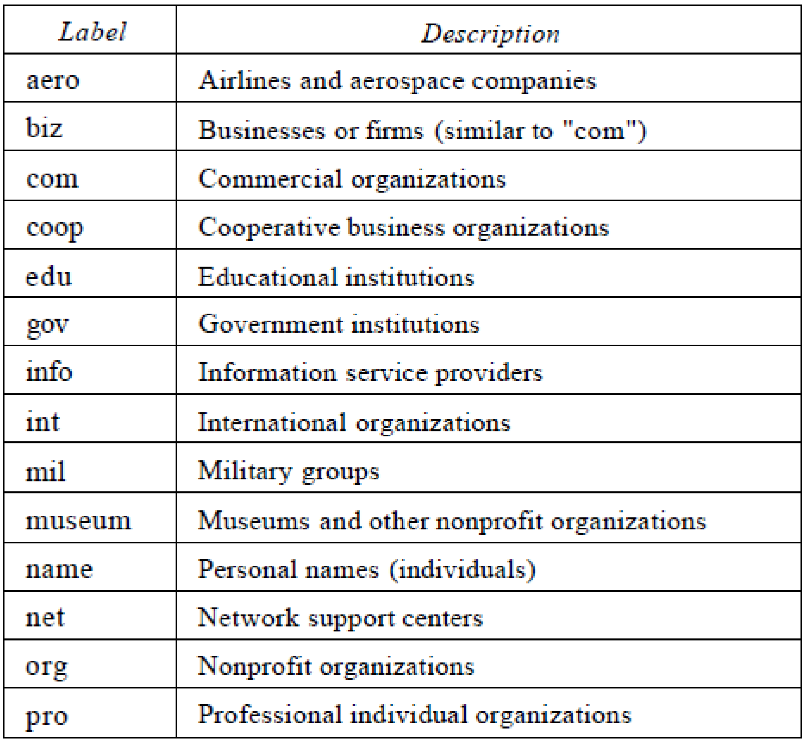


DNS used in the Internet.

Generic Domains

The **generic domains** define registered hosts according to their generic behavior. Each node in the tree defines a domain, which is an index to the domain name space database.

Looking at the tree, we see that the first level in the generic domains section allows 14 possible labels. These labels describe the organization types as listed in Table.



Label	*Description*
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to "com")
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International organizations
mil	Military groups
museum	Museums and other nonprofit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

Country Domains

The country domains section uses two-character country abbreviations (e.g., us for United States). Second labels can be organizational, or they can be more specific, national designations. The United States, for example, uses state abbreviations as a subdivision of us (e.g., ca.us.). Figure shows the country domains section. The address *anza.cup.ca.us* can be translated to De Anza College in Cupertino, California, in the United States.

Inverse Domain

The inverse domain is used to map an address to a name. This may happen, for example, when a server has received a request from a client to do a task. Although the server has a file that contains a list of authorized clients, only the IP address of the client (extracted from the received IP packet) is listed. The server asks its resolver to send a query to the DNS server to map an address to a name to determine if the client is on the authorized list.

This type of query is called an inverse or pointer (PTR) query. To handle a pointer query, the inverse domain is added to the domain name space with the first-level node called *arpa* (for historical reasons). The second level is also one single node named *in-addr* (for inverse address). The rest of the domain defines IP addresses. The servers that handle the inverse domain are also hierarchical. This means the netid part of the address should be at a higher level than the subnetid part, and the subnetid part higher than the host id part. In this way, a server serving the whole site is at a higher level than the servers serving each subnet. This configuration makes the domain look inverted when compared to a generic or country domain. To follow the convention of reading the domain labels from the bottom to the top, an IF address such as 132.34.45.121 (a class B address with net id 132.34) is read as 121.45.34.132.in-addr. arpa. See Figure for an illustration of the inverse domain configuration.

OSPF

The Open Shortest Path First or OSPF protocol is an intradomain routing protocol based on link state routing. Its domain is also an autonomous system. Areas To handle routing efficiently and in a timely manner, OSPF divides an autonomous system into areas. An area is a collection of networks, hosts, and routers all contained within an autonomous system. An autonomous system can be divided into many different areas. All networks inside an area must be connected. Routers inside an area flood the area with routing information. At the border of an area, special routers called area border routers summarize the information about the area and send it to other areas. Among the areas inside an autonomous system is a special area called the backbone; all the areas inside an autonomous system must be connected to the backbone. In other words, the backbone serves as a primary area and the other areas as secondary areas. This does not mean that the routers within areas cannot be connected to each other, however. The routers inside the backbone are called the backbone routers. Note that a backbone router can also be an area border router.

If, because of some problem, the connectivity between a backbone and an area is broken, a virtual link between routers must be created by an administrator to allow continuity of the functions of the backbone as the primary area. Each area has an area identification. The area identification of the backbone is zero.

Metric: The OSPF protocol allows the administrator to assign a cost, called the metric, to each route. The metric can be based on a type of service (minimum delay, maximum throughput, and so on). As a matter of fact, a router can have multiple routing tables, each based on a different type of service.

BGP

Border Gateway Protocol (BGP) is an interdomain routing protocol using path vector routing. It first appeared in 1989 and has gone through four versions. BGP Sessions: The exchange of routing information between two routers using BGP takes place in a session. A session is a connection that is established between two BGP routers only for the sake of exchanging routing information. To create a reliable environment, BGP uses the services of TCP. In other words, a session at the BGP level, as an application program, is a connection at the TCP level. However, there is a subtle difference between a connection in TCP made for BGP and other application programs. When a TCP connection is created for BGP, it can last for a long time, until something unusual happens.

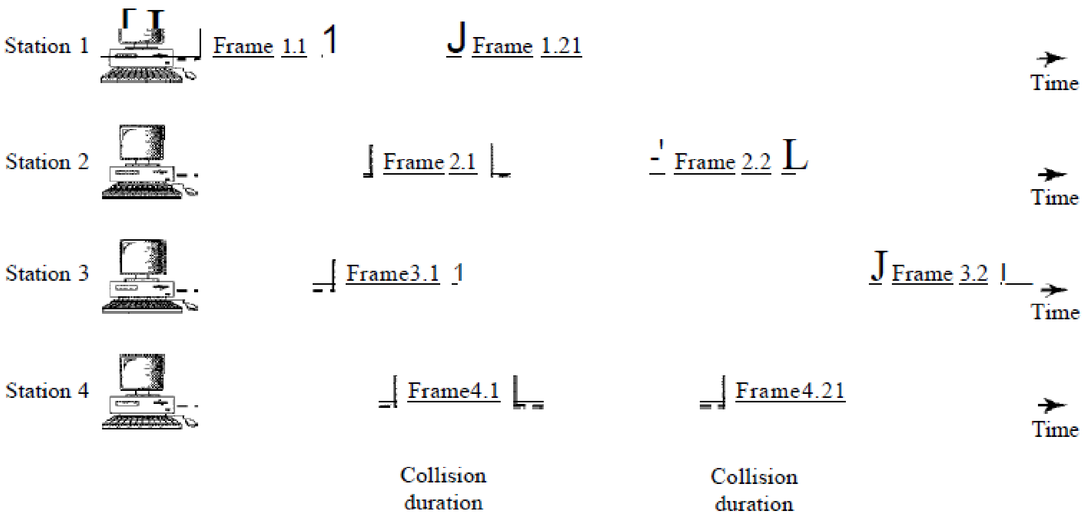
ALOHA

ALOHA, the earliest random access method, was developed at the University of Hawaii in early 1970. It was designed for a radio (wireless) LAN, but it can be used on any shared medium.

It is obvious that there are potential collisions in this arrangement. The medium is shared between the stations. When a station sends data, another station may attempt to do so at the same time. The data from the two stations collide and become garbled.

Pure ALOHA

The original ALOHA protocol is called pure ALOHA. This is a simple, but elegant protocol. The idea is that each station sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is the possibility of collision between frames from different stations. Figure 12.3 shows an example of frame collisions
in pure ALOHA.



There are four stations (unrealistic assumption) that contend with one another for access to the shared channel. The figure shows that each station sends two frames; there are a total of eight frames on the shared medium. Some of these frames collide because multiple frames are in contention for the shared channel. Figure 12.3 shows that only There are four stations (unrealistic assumption) that contend with one another for access to the shared channel. The figure shows that each station sends two frames; there are a total of eight frames on the shared medium. Some of these frames collide because multiple frames are in contention for the shared channel. Figure 12.3 shows that only two frames survive: frame 1.1 from station 1 and frame 3.2 from station 3. We need to mention that even if one bit of a frame coexists on the channel with one bit from another frame, there is a collision and both will be destroyed.

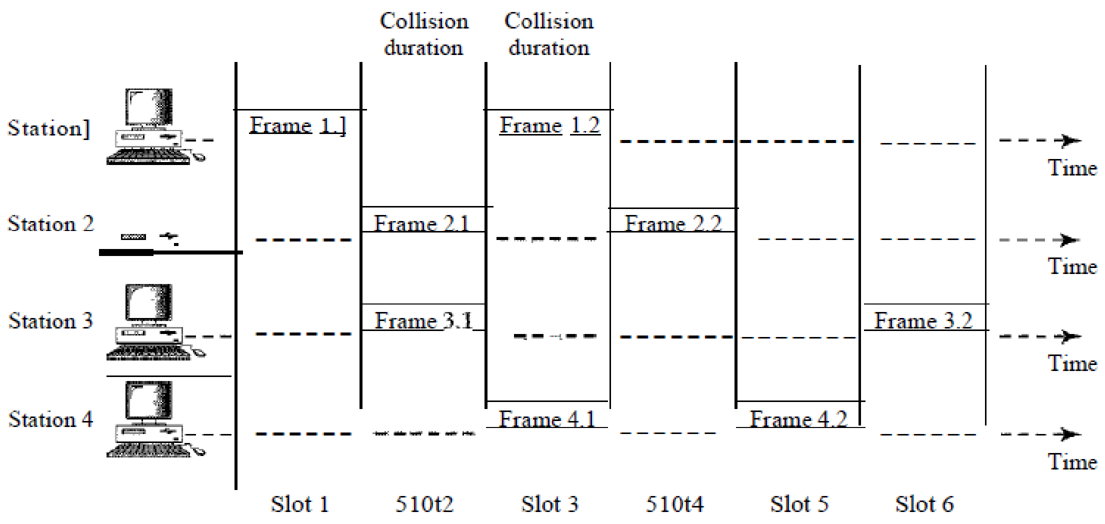
It is obvious that we need to resend the frames that have been destroyed during transmission. The pure ALOHA protocol relies on acknowledgments from the receiver. When a station sends a frame, it expects the receiver to send an acknowledgment. If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame.

A collision involves two or more stations. If all these stations try to resend their frames after the time-out, the frames will collide again. Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame. The randomness will help avoid more collisions. We call this time the back-off time *TB.*

Slotted ALOHA

Pure ALOHA has a vulnerable time of 2 x *Tfr* . This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or soon before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA.

In slotted ALOHA we divide the time into slots of Tfr s and force the station to send only at the beginning of the time slot. Figure 12.6 shows an example of frame collisions in slotted ALOHA.



Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. Of course, there is still the possibility of collision if two stations try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to *Tfr.*