UNIT- V

Transport layer UDP, TCP. Connection establishment and termination, sliding window, flow and congestion control, timers, retransmission, TCP extensions, Queuing theory, Single and multiple server queuing models, Little's formula. **Application Layer**. Network Application services and protocols including e-mail, www, DNS, SMTP, IMAP, FTP, TFTP, Telnet, BOOTP, HTTP, IPSec, Firewalls.

TRANSMISSION CONTROL PROTOCOL

Transmission Control Protocol (TCP) is a connection-oriented, reliable protocol. TCP explicitly defines connection establishment, data transfer, and connection teardown phases to provide a connection-oriented service.

TCP Services

Process-to-Process Communication

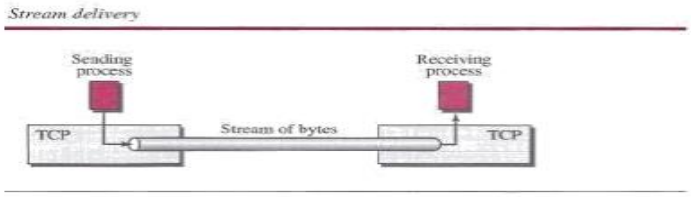
As with UDP, TCP provides process-to-process communication using port numbers. We have already given some of the port numbers used by TCP.

Stream Delivery Service

In UDP, a process sends messages with predefined boundaries to UDP for delivery. UDP adds its own header to each of these messages and delivers it to IP for transmission.

TCP, on the other hand, allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.

TCP creates an environment in which the two processes seem to be connected by an imaginary "tube" that carries their bytes across the Internet.



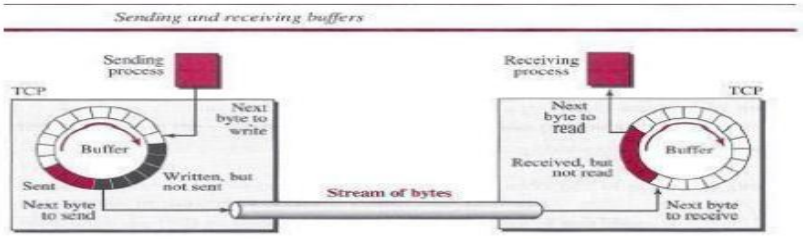
Sending and Receiving Buffers

Because the sending and the receiving processes may not necessarily write or read data at the same rate, TCP needs buffers for storage.

There are two buffers, the sending buffer and the receiving buffer, one for each direction.

- At the sender, the buffer has three types of chambers. The white section contains empty chambers that can be filled by the sending process (producer).
- The colored area holds bytes that have been sent but not yet acknowledged.
- The TCP sender keeps these bytes in the buffer until it receives an acknowledgment. The shaded area contains bytes to be sent by the sending TCP.
- The operation of the buffer at the receiver is simpler. The circular buffer is divided into two areas (shown as white and colored).

- The operation of the buffer at the receiver is simpler. The circular buffer is divided into two areas (shown as white and colored).
- The white area contains empty chambers to be filled by bytes received from the network.
- The colored sections contain received bytes that can be read by the receiving process. When a byte is read by the receiving process, the chamber is recycled and added to the pool of empty chambers.

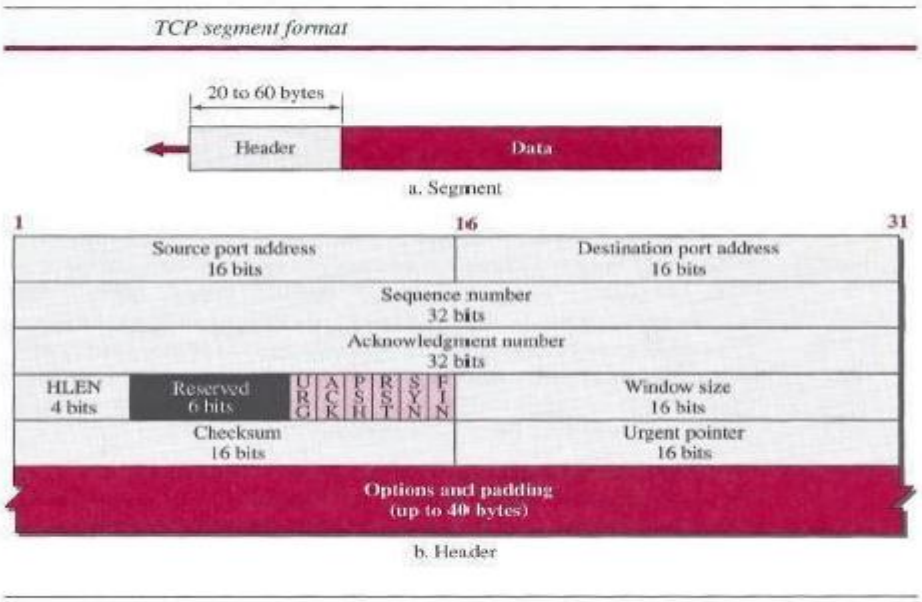


Segments

- Although buffering handles the disparity between the speed of the producing and consuming Processes, we need one more step before we can send data.
- The network layer, as a service provider for TCP, needs to send data in packets, not as a stream of bytes. At the transport layer, TCP groups a number of bytes together into a packet called a *segment*.
- The segments are encapsulated in an IP datagram and transmitted. This entire operation is transparent to the receiving process.

Format

The segment consists of a header of 20 to 60 bytes, followed by data from the application program.
The header is 20 bytes if there are no options and up to 60 bytes if it contains options.



Source port address This is a 16-bit field that defines the port number of the application program in the host that is sending the segment.

Destination port address This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment.

Sequence number This 32-bit field defines the number assigned to the first byte of data contained in this segment.

Acknowledgment number This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party.

Header length This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes.

A TCP Connection

- TCP is connection-oriented. a connection-oriented transport protocol establishes a logical path between the source and destination.
- All of the segments belonging to a message are then sent over this logical path.
- TCP operates at a higher level. TCP uses the services of IP to deliver individual segments to the receiver, but it controls the connection itself.
- In TCP, connection-oriented transmission requires three phases: connection establishment, data transfer, and connection termination.

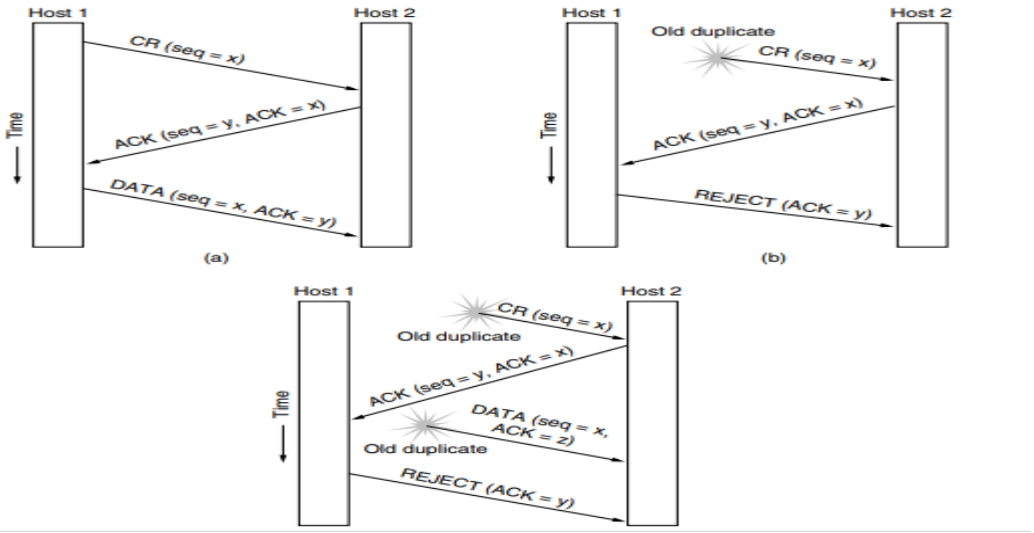


Figure 6-11. Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST. (a) Normal operation. (b) Old duplicate CONNECTION REQUEST appearing out of nowhere. (c) Duplicate CONNECTION REQUEST and duplicate ACK.

Connection Establishment

TCP transmits data in full-duplex mode. When two TCPs in two machines are connected, they are able to send segments to each other simultaneously.

Three- Way Handshaking

The connection establishment in TCP is called *three-way handshaking.* an application program, called the *client,* wants to make a connection with another application program, called the *server,*

using TCP as the transport-layer protocol The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This request is called a *passive open.*

Although the server TCP is ready to accept a connection from any machine in the world, it cannot make the connection itself.

The client program issues a request for an *active open.* A client that wishes to connect to an open server tells its TCP to connect to a particular server.

The r

Connection Release

Releasing a connection is easier than establishing one. Nevertheless, there are more pitfalls than one might expect here. As we mentioned earlier, there are two styles of terminating a connection: asymmetric release and symmetric release.

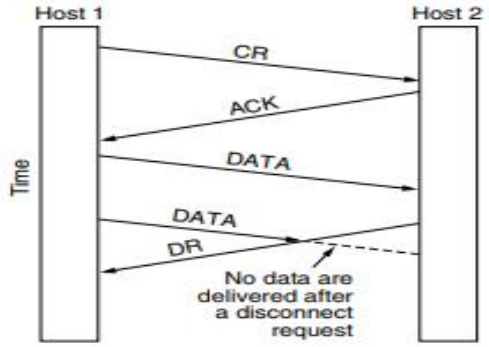


Figure 6-12. Abrupt disconnection with loss of data.

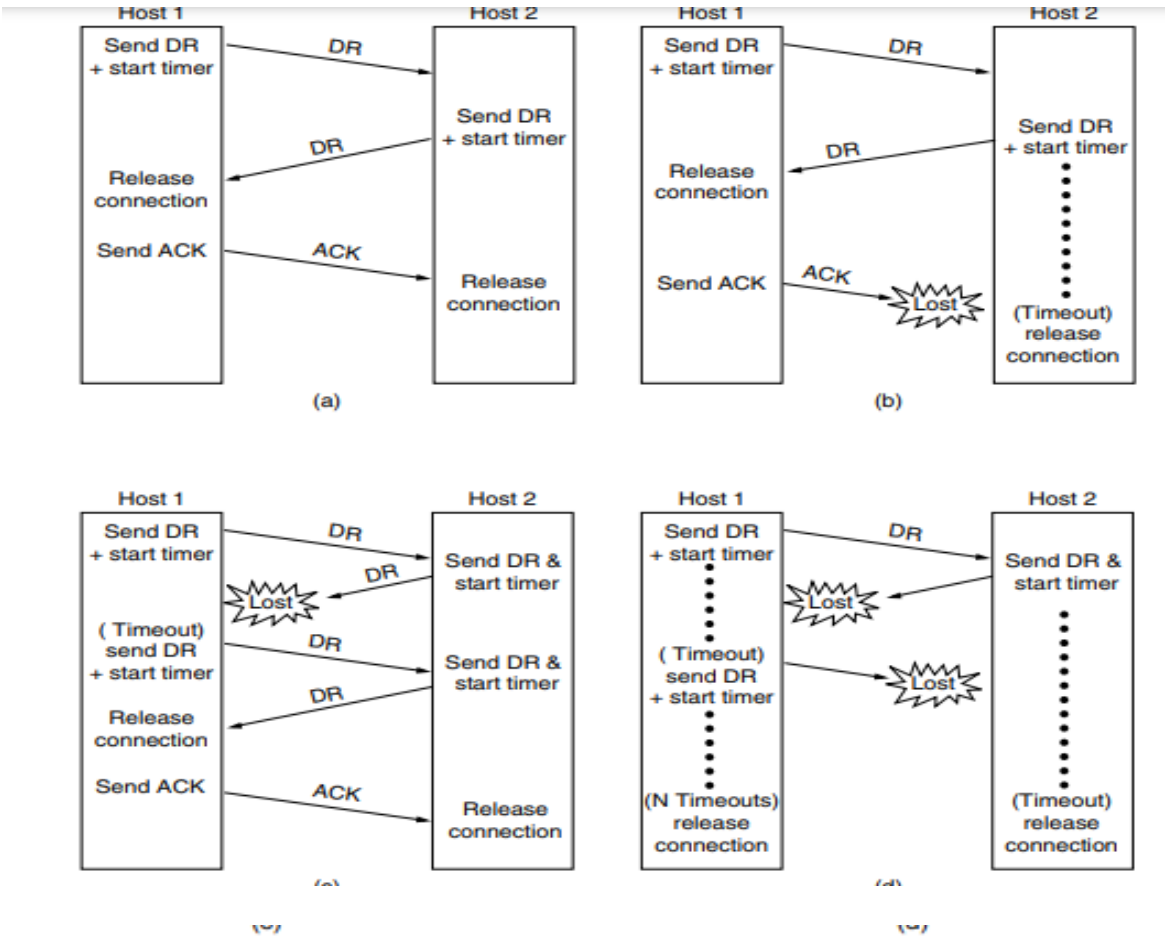


Figure 6-14. Four protocol scenarios for releasing a connection. (a) Normal case of three-way handshake. (b) Final ACK lost. (c) Response lost. (d) Response lost and subsequent DRs lost.

UDP Protocol

UDP provides connectionless, unreliable, datagram service. Connectionless service means that there is no logical connection between the two ends exchanging messages. Each message is an independent entity encapsulated in a datagram.

UDP does not see any relation (connection) between consequent datagram coming from the same source and going to the same destination.

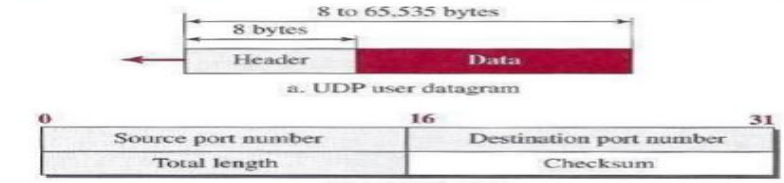
UDP has an advantage: it is message-oriented. It gives boundaries to the messages exchanged. An application program may be designed to use UDP if it is sending small messages and the simplicity and speed is more important for the application than reliability.

User Datagram

UDP packets, called *user datagram,* have a fixed-size header of 8 bytes made of four fields, each of 2 bytes (16 bits).

. The 16 bits can define a total length of 0 to 65,535 bytes. However, the total length needs to be less because a UDP user datagram is stored in an IP datagram with the total length of 65,535 bytes. The last field can carry the optional checksum

User datagram packet format



UDP Services

Process-to-Process Communication

UDP provides process-to-process communication using **socket addresses,** a combination of **IP** addresses and port numbers.

Connectionless Services

As mentioned previously, UDP provides a *connection less service.* This means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user data grams even if they are coming from the same source process and going to the same destination program.

Flow Control

UDP is a very simple protocol. There is no *flow control,* and hence no window mechanism. The receiver may overflow with incoming messages.

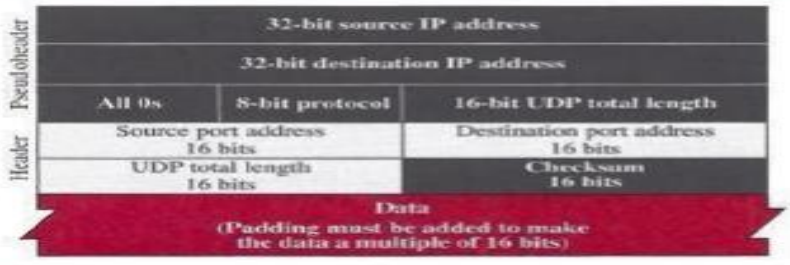
Error Control

There is no *error control* mechanism in UDP except for the checksum. This means that the sender does not know if a message has been lost or duplicated.

Checksum

UDP checksum calculation includes three sections: a pseudo header, the UDP header, and the data coming from the application layer. The *pseudo header* is the part of the header of the IP packet in which the user datagram is to be encapsulated with some fields filled with 0s

Pseudoheader for checksum calculation



UDP Applications

UDP Features

Connectionless Service

As we mentioned previously,

- UDP is a connectionless protocol. Each UDP packet is independent from other packets sent by the same application program. This feature can be considered as an advantage or disadvanta8e depending on the application requirements.

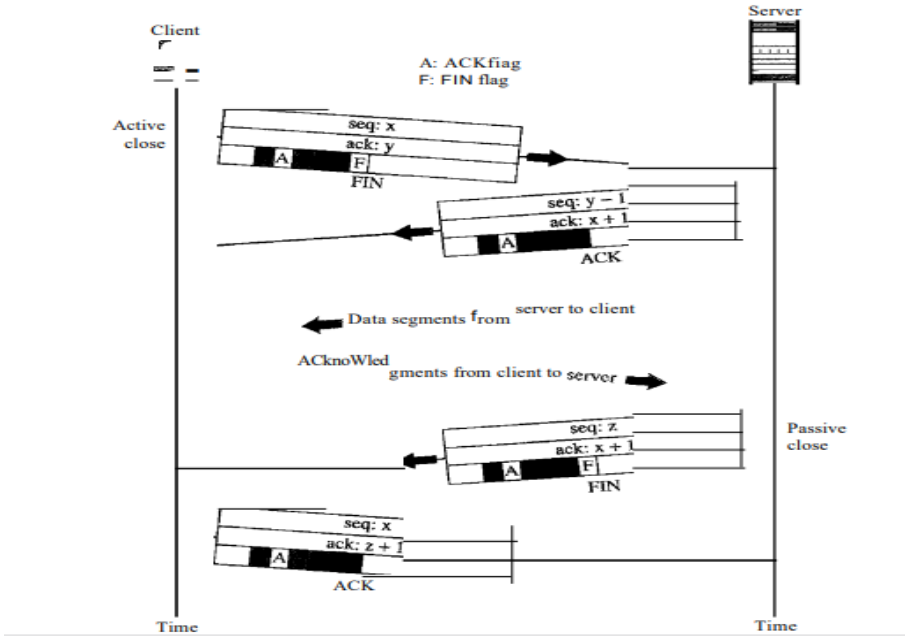
- UDP does not provide error control; it provides an unreliable service. Most applications expect reliable service from a transport-layer protocol. Although a reliable service is desirable.

Transmission control protocol (TCP)	User datagram protocol (UDP)
TCP is a connection-oriented protocol. Connection-orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data.	UDP is the Datagram oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, and terminating a connection. UDP is efficient for broadcast and multicast type of network transmission.
TCP is reliable as it guarantees the delivery of data to the destination router.	The delivery of data to the destination cannot be guaranteed in UDP.
TCP provides extensive error checking mechanisms. It is because it provides flow control and acknowledgement of data.	UDP has only the basic error checking mechanism using checksums.
Acknowledgement segment is present.	No acknowledgement segment.
Sequencing of data is a feature of Transmission Control Protocol (TCP). this means that packets arrive in-order at the receiver.	There is no sequencing of data in UDP. If the order is required, it has to be managed by the application layer.
TCP is comparatively slower than UDP.	UDP is faster, simpler, and more efficient than TCP.
Retransmission of lost packets is possible in TCP, but not in UDP.	There is no retransmission of lost packets in the User Datagram Protocol (UDP).
TCP has a (20-60) bytes variable length header.	UDP has an 8 bytes fixed-length header.
TCP is heavy-weight.	UDP is lightweight.

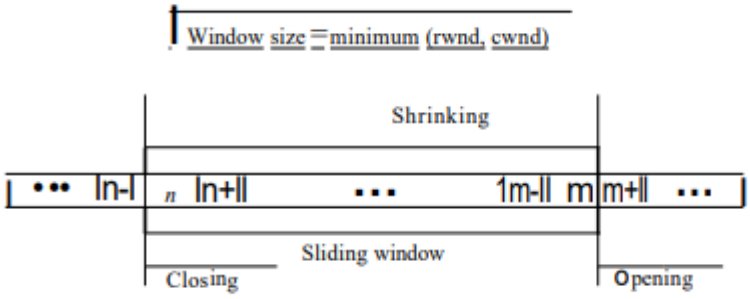
Transmission control protocol (TCP)	User datagram protocol (UDP)
Uses handshakes such as SYN, ACK, SYN-ACK	It's a connectionless protocol i.e. No handshake
TCP doesn't support Broadcasting.	UDP supports Broadcasting.
TCP is used by HTTP, HTTPs, FTP, SMTP and Telnet.	UDP is used by DNS, DHCP, TFTP, SNMP, RIP, and VoIP.

FLOW CONTROL & ERROR CONTROL SLIDING WINDOW PROTOCOL

TCP uses a sliding window, The sliding window protocol used by TCP, however, is something between the Go-Back-N and Selective Repeat sliding window



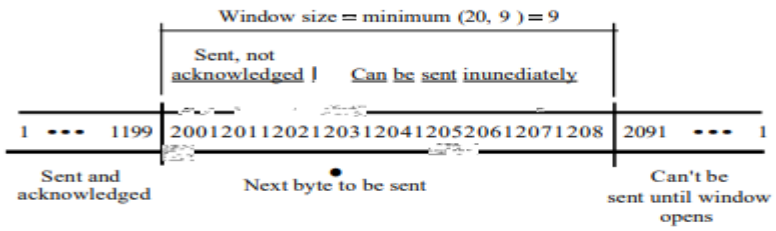
Sliding window



A sliding window is used to make transmission more efficient as weD as to control the flow of data so that the destination does not become overwhelmed with data. TCP sliding windows are byte-oriented.

The size of the window at one end is determined by the lesser of two values: *window (rwnd)* or *congestion window (cwnd)*. The *receiver window* is the val tised by the opposite end in a segment containing acknowledgment. It is the n bytes the other end can accept before its buffer overflows and data are discar congestion window is a value determined by the network to avoid congestion discuss congestion later in the chapter.

Example 23.6



Some points about TCP sliding windows:

- The size of the window is the lesser of *rwnd* and *cwnd.*
- The source does not have to send a full window's worth of data.
- The window can be opened or closed by the receiver, but should not be shrunk.
- The destination can send an acknowledgment at any time as long as it does not result in a shrinking window.
- The receiver can temporarily shut down the window; the sender, however, can always send a segment of 1 byte after the window is shut down.

Error Control

TCP is a reliable transport layer protocol. This means that an application program that delivers a stream of data to TCP relies on TCP to deliver the entire stream to the application program on the other end in order, without error, and without any part lost or duplicated.

TCP provides reliability using error control. Error control includes mechanisms for detecting corrupted segments, lost segments, out-of-order segments, and duplicated segments. Error control also includes a mechanism for correcting errors after they are detected. Error detection and correction in TCP is achieved through the use of three simple tools: checksum, acknowledgment, and time-out.

Checksum

Each segment includes a checksum field which is used to check for a corrupted segment.

Acknowledgment

TCP uses acknowledgments to confirm the receipt of data segments. Control segments

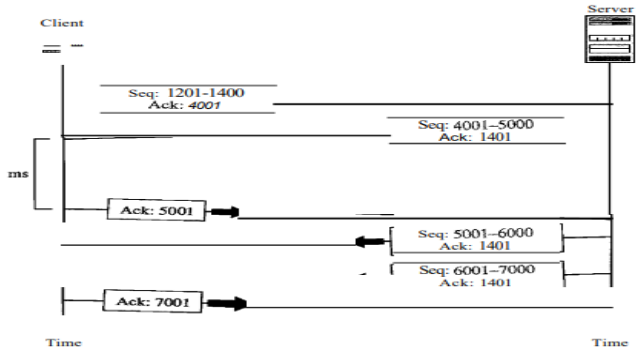
Retransmission

The heart of the error control mechanism is the retransmission of segments.

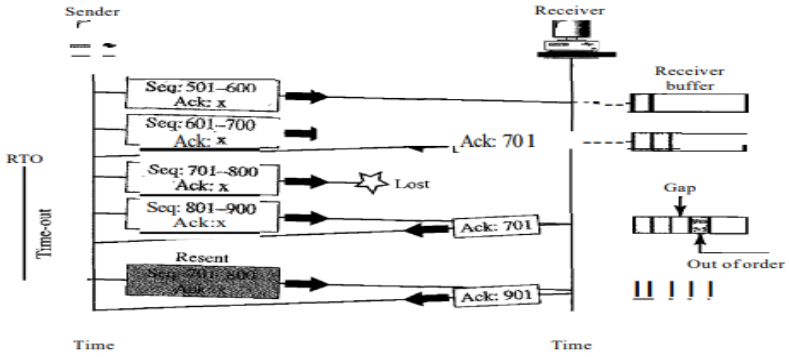
In modern implementations, a retransmission occurs if the retransmission timer expires or three duplicate ACK segments have arrived.

No retransmission timer is set for an ACK segment.

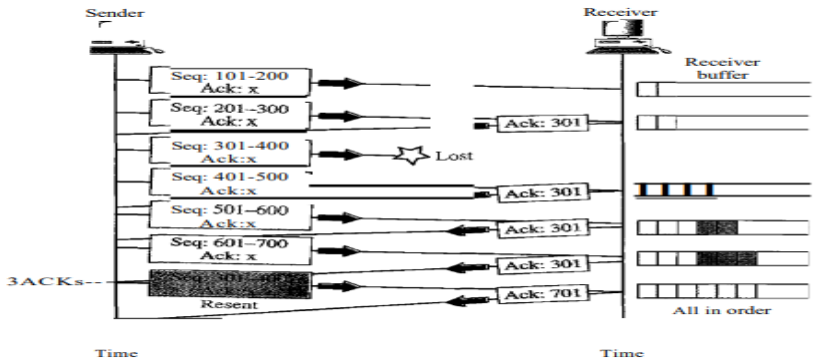
Nonnaloperation



Lost segment



Fast Retransmission



APPLICATION LAYER

Responsibilities of the application layer are as follows :

» **Network abstraction** : The application layer provides an abstraction of the underlaying network to an end user and an application.

» **File access and transfer** : It allows a use to access, download or upload files from/to a remote host.

» **Mail services** : It allows the users to use the mail services.

» **Remote login** : It allows logging into a host which is remote

» **World Wide Web (WWW)** : Accessing the Web pages is also a part of this layer.

ELECTRONIC MAIL

Electronic mail (or e-mail) allows users to exchange messages.

- In an application such as HTTP or FTP, the server program is running all the time, waiting for a
- request from a client.
- When the request arrives, the server provides the service. There is a request and there is a response.
- In the case of electronic mail, the situation is different. First, e-mail is considered a one-way transaction
- The users run only client programs when they want and the intermediate servers apply the client/server paradigm.
- In the common scenario, the sender and the receiver of the e-mail, are connected via a LAN or a WAN to two mail servers.
- The administrator has created one mailbox for each user where the received messages are stored
- A *mailbox* is part of a server hard drive, a special file with permission restrictions.
- Only the owner of the mailbox has access to it. The administrator has also created a queue (spool) to store messages waiting to be sent.
- A simple e-mail use three different *agents:* a user agent (UA), a message transfer agent (MTA), and a message access agent (MAA).

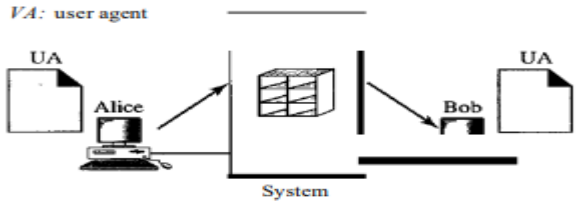
E-M ail One of the most popular Internet services is electronic mail (e-mail).It is exchange of computer stored messages by telecommunication. Email messages are usually encoded in ASCII text ,non text files.

Architecture

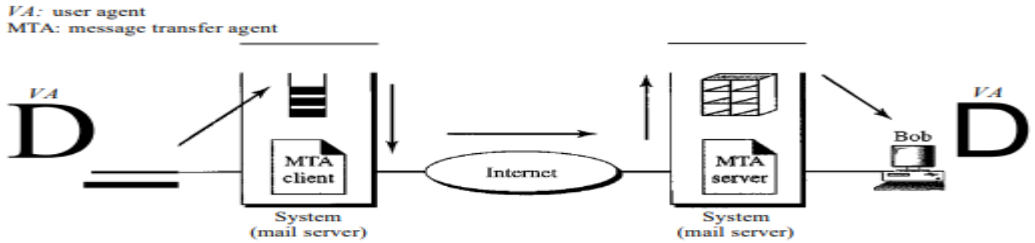
To explain the architecture of e-mail, we give four scenarios

First Scenario

When the sender and the receiver of an e-mail are on the same system, we need only two user agents.

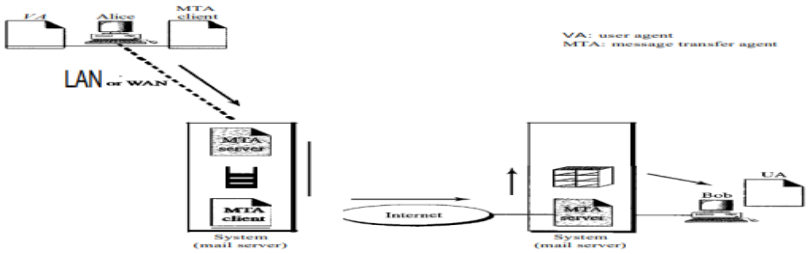


Second Scenario



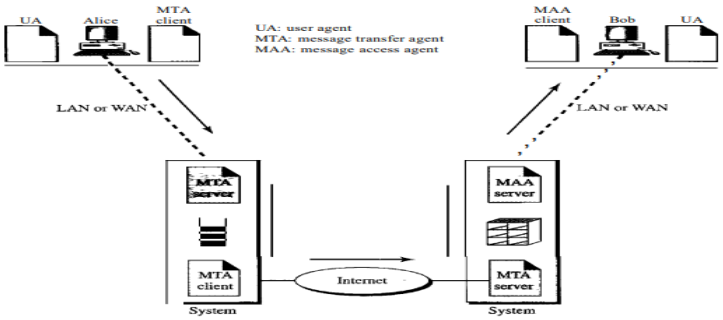
When the sender and the receiver of an e-mail are on different systems, we need two VAs and a pair of MTAs (client and server).

Third Scenario



When the sender is connected to the mail server via a LAN or a WAN, we need two *VAs* and two pairs of MTAs (client and server).

Fourth Scenario



When both sender and receiver are connected to the mail server via a LAN or a WAN, we need two VAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server). This is the most common situation today.

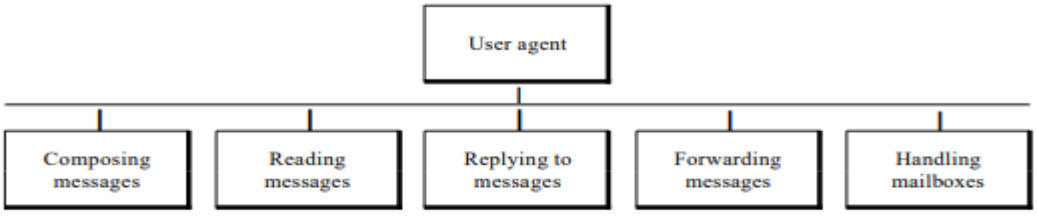
COMPONENTS OF E-MAIL

User Agent

The first component of an electronic mail system is the user agent *(VA)*. It provides service to the user to make the process of sending and receiving a message easier.

Services Provided by a User Agent

A user agent is a software package (program) that composes, reads, replies to, and forwards messages. It also handles mailboxes. Figure 26.11 shows the services of a typical user agent.



Composing Messages A user agent helps the user compose the e-mail message to be sent out. Most user agents provide a template on the screen to be filled in by the user.

Reading Messages The second duty of the user agent is to read the incoming messages. When a user invokes a user agent, it first checks the mail in the incoming mailbox. Most user agents show a one-line summary of each received mail. Each e-mail contains the following fields.

1. A number field.
2. A flag field that shows the status of the mail such as new, already read but not replied to, or read and replied to.
3. The size of the message.
4. The sender.
5. The optional subject field.

Replying to Messages After reading a message, a user can use the user agent to reply to a message. A user agent usually allows the user to reply to the original sender or to

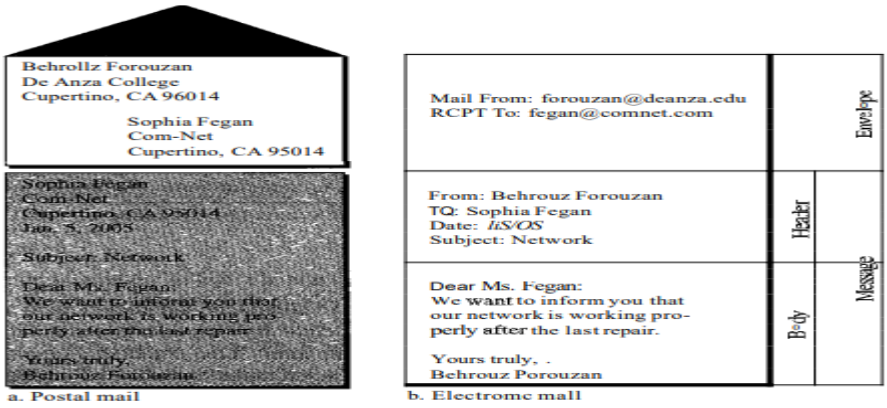
Forwarding Messages *Replying* is defined as sending a message to the sender or recipients of the copy. *Forwarding* is defined as sending the message to a third party. A

Handling Mailboxes

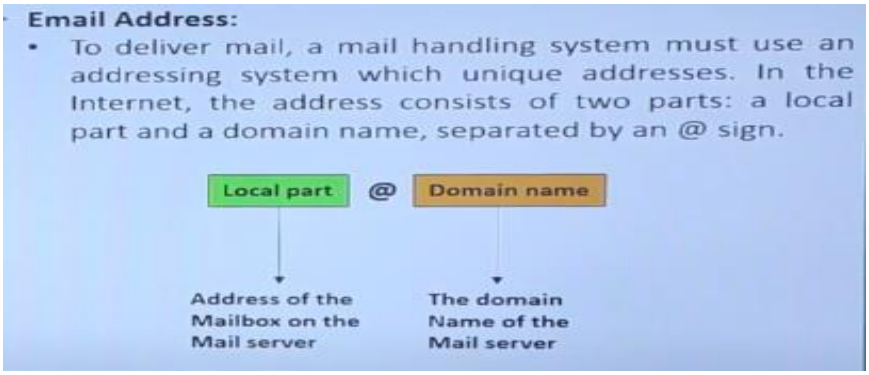
A user agent normally creates two mailboxes: an inbox and an outbox. Each box is a file with a special format that can be handled by the user agent. The inbox keeps all the received e-mails until they are deleted by the user. The outbox keeps all the sent e-mails until the user deletes them. Most user agents today are capable of creating customized mailboxes.

User Agent Types

There are two types of user agents: command-driven and GUI-based.

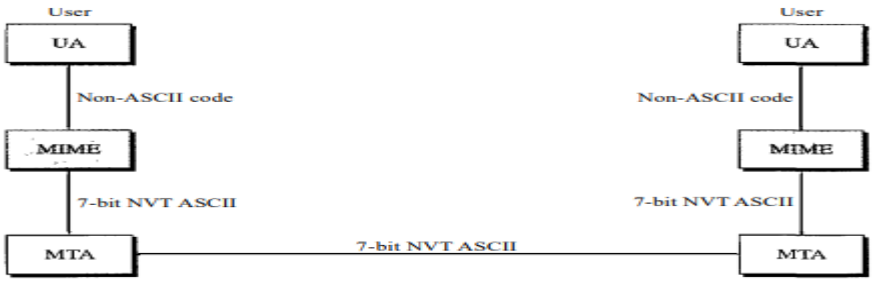


a. Postal mail b. Electromc mail



MIME

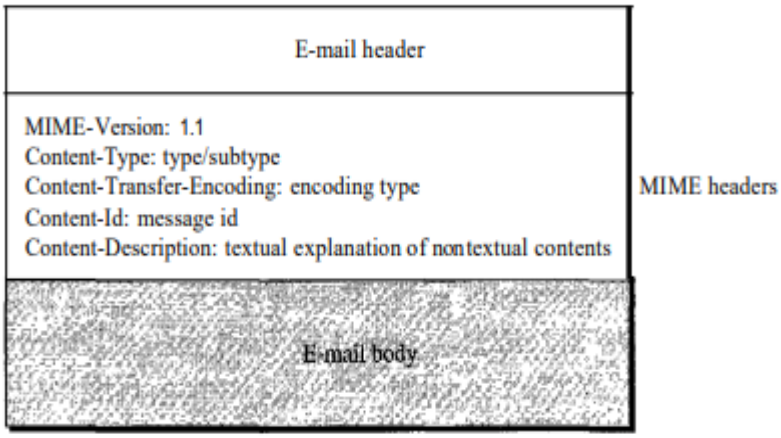
Multipurpose Internet Mail Extensions (MIME) is a supplementary protocol that allows non-ASCII data to be sent through e-mail. MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers them to the client MTA to be sent through the Internet. The message at the receiving side is transformed back to the original data.



MIME defines five headers that can be added to the original e-mail header section to define the transformation parameters:

1. MIME-Version
2. Content-Type
3. Content-Transfer-Encoding
4. Content-Id
5. Content-Description

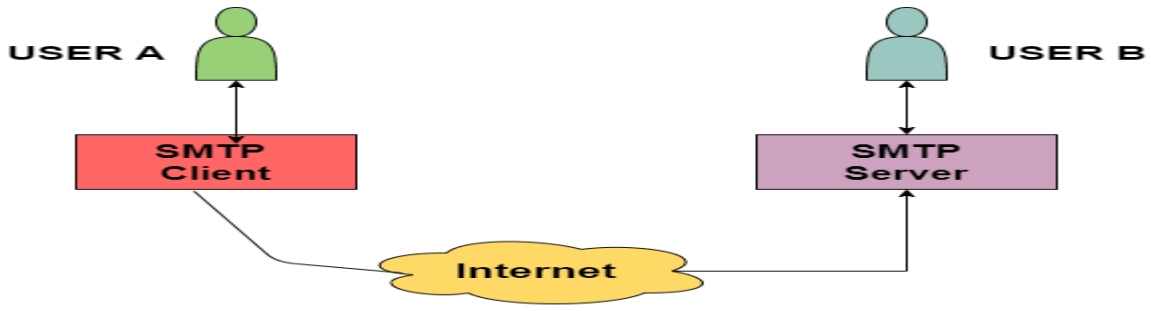
MIME header



Message Transfer Agent: SMTP

SMTP mainly stands for Simple Mail Transfer Protocol. Basically, the actual transfer of mail is done through the message transfer agents(MTA). Thus in order to send the mail, the system must have the **client MTA** and in order to receive the mail, the system must have a server MTA.

- In order to define the **MTA client** and **server** on the Internet, there is a formal way and it is known as **Simple Mail Transfer Protocol(SMTP)**.
- SMTP also makes the use of TCP/IP for sending and receiving e-mail.
- SMTP is based on the client/server model.
- The original standard port for SMTP is Port 25.
- Using this protocol, the client who wants to send the e-mail first opens a TCP connection to the SMTP server and then sends the e-mail across the TCP connection. It is important to note that the SMTP server is always in listening mode. As soon as it listens for the TCP connection from any client then the connection is Initiated on port 25 and after the successful connection, the client sends the e-mail/message immediately.



SMTP is used two times while sending an Email:

1. Between the Sender and Sender's mail server
2. Between the Sender's mail server and the Receiver's mail server

It is important to note that in order to receive or download the email,

- There is a need for another protocol between the mail server of receiver and the receiver.
- Commonly used protocols are POP3 and IMAP. Thus these two are mail access agents.

Architecture of SMTP

All the users make use of **User Agent (UA)**. The Mail Transfer Agent **(MTA)** mainly helps to exchange all the messages in between both sender and receiver using the TCP/IP. The system administrator has the authority to configure the set up of local MTA, thus the users who are sending the email do not need to deal with the MTA.

The MTA keeps the queue in the pool of messages, if the receiver is not available at that moment then MTA can schedule the repeat delivery of all the messages.

MTA (Mail User Agent) forwards the emails into mailboxes of the user's local system, and then the user agent (UA) can download those messages at any time.

The SMTP Client as well as the SMTP server both has two main components and these are:

- UA(User-Agent)
- MTA(Mail Transfer Agent)

Let us now take a look at communication between the sender and the receiver:

The user agent at the sender side prepares the message and then sent it to the MTA. The task of the MTA is to transfer the Email across the network to the Receiver MTA. Also in order to send the Email, a system must have the client MTA and in order to receive the email, a system must have a server MTA.

Sending the Email

An email is sent between the sender and receiver using a series of request and response messages. An Email mainly consists of two parts **a header and body**. The body part of an email indicates the main message area. It is the actual information that is to be read by the receiver. The header mainly contains the address of the sender and recipient and it also contains the subject of the email.

In order to terminate the header of the email, there is a NULL line, everything after the NULL line is considered as the body of the message.

Receiving the Email

Mailboxes are checked by the user agent at the server side at a particular interval of time. In case if any information is received then it informs the receiver about the email.

At the time when the user tries to read the email then MTA mainly displays a list of emails with their short description in the mailbox. If the user selects any of the emails then can easily view the contents inside the email.

SMTP Protocol Method

1. **Store-and-Forward Method** The store and forward method is used within an organization.
2. **End-to-End Method** Mainly the end-to-end method is used to communicate between the different organizations

An SMTP client is the one who wants to send the mail and will definitely contact the destination's host SMTP directly in the order to send the Email to the destination. Also, the session is initiated by the client SMPT.

On the other hand, the SMTP server will keep the mail to itself until it is successfully copied to the SMTP at the receiver. The server SMTP mainly responds to the session request.

Thus the session is started by the client-SMTP and the server-SMTP will respond to the request of the sender.

Characteristics of SMTP

Let us take a look at the characteristics of the SMTP:

- SMTP makes use of Port 25.
- It makes use of persistent TCP connections and thus can send multiple emails all at once.
- It is a stateless protocol.
- It is a connection-oriented protocol.
- It makes use of TCP at the transport layer.
- It is a push control protocol.

Advantages of SMTP

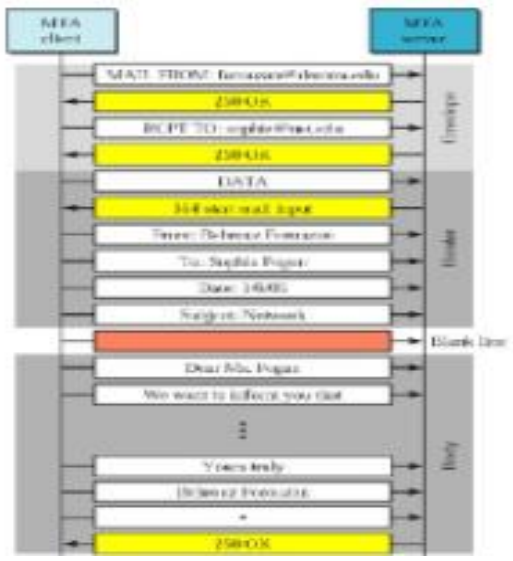
Let us take a look at the advantages offered by the simple mail transfer protocol(SMTP):

- SMTP offers reliability in terms of the outgoing email messages.
- It is the simplest form of communication between various computers in a network via Email.
- In those cases where a particular message was not **delivered successfully** then, the SMTP server always tries to re-send the same message until the **transmission** becomes **successful**.

Disadvantages of SMTP

- SMTP does not provide good security.
- It is only limited to 7-bit ASCII characters.

- Beyond some specific length, email messages are rejected by SMTP servers.
- The usefulness of SMTP is limited by its simplicity.
- With the help of SMTP, the transmission of executable files and binary files is not possible until they get converted into text files.



Internet Message Access Protocol (IMAP) is an application layer protocol that operates as a contract for receiving emails from the mail server. It was designed by Mark Crispin in 1986 as a remote access mailbox protocol, the current version of IMAP is IMAP4. It is used as the most commonly used protocol for retrieving emails. This term is also known as Internet mail access protocol, Interactive mail access protocol, and Interim mail access protocol.

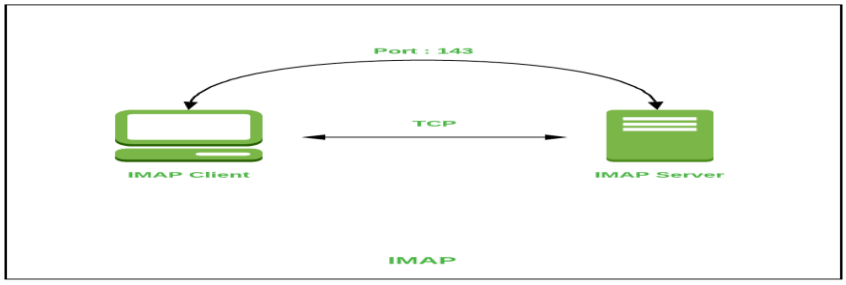
Features of IMAP :

- It is capable of managing multiple mailboxes and organizing them into various categories.
- Provides adding of message flags to keep track of which messages are being seen.
- It is capable of deciding whether to retrieve email from a mail server before downloading.
- It makes it easy to download media when multiple files are attached.

Working of IMAP :

IMAP follows Client-server Architecture and is the most commonly used email protocol. It is a combination of client and server process running on other computers that are connected through a network. This protocol resides over the TCP/IP protocol for communication. Once the communication is set up the server listens on port 143 by default which is non-encrypted. For the secure encrypted communication port, 993 is used.

Architecture of IMAP :



Advantages :

- It offers synchronization across all the maintained sessions by the user.
- It provides security over POP3 protocol as the email only exists on the IMAP server.
- Users have remote access to all the contents.
- It offers easy migration between the devices as it is synchronized by a centralized server.
- There is no need to physically allocate any storage to save contents.

Disadvantages :

- IMAP is complex to maintain.
- Emails of the user are only available when there is an internet connection.
- It is slower to load messages.
- Some emails don't support IMAP which makes it difficult to manage.
- Many browser-based solutions are unavailable due to not support of IMAP.

POP Protocol

In this tutorial, we will be covering the concept of POP which is another Application layer protocol.

POP is a short form of Post Office Protocol. It is another protocol present at the Application Layer of the OSI reference model.

- POP is mainly a message access protocol.
- POP is basically an internet standard protocol and as we already told you it works on the application layer and is used by the local email software in order to retrieve emails from the remote email server over the TCP/IP connection.
- The Post office Protocol (POP) does not allow any search facility.
- This protocol mainly allows one protocol to be created on the server.
- As this protocol supports offline access to the messages and so less internet usage time is required by this.
- Non-email data is not accessed by this protocol.
- Some of the common clients that make use of POP3 are Gmail, Netscape, Internet Explorer, Eudora.

History of POP

The POP(post office protocol) was published in 1984 by Internet Engineering Task Force. After that, it has been updated two times, because the backend developers want to make the layout simple.

The second version of POP was developed in 1985 and known as POP2 and this version needs the SMTP protocol in order to push the emails.

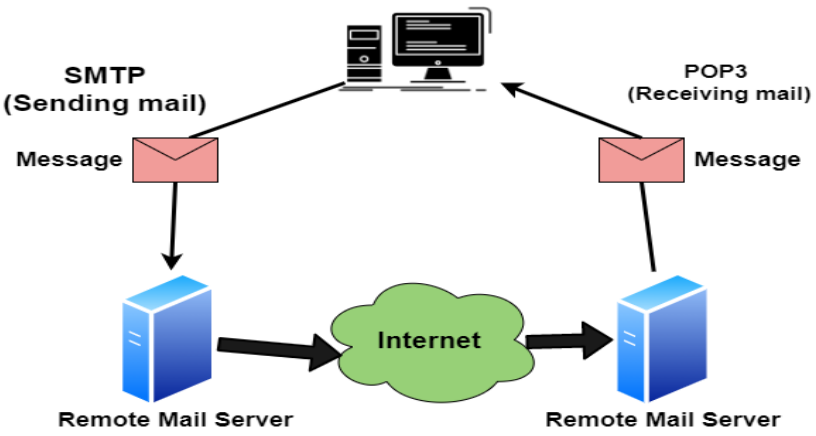
Then after the third version of POP was released in 1988 and known as POP3, this version does not require the SMTP protocol. The **POP(Post office protocol version 3)** is also integrated into famous e-mail software, like Eudora and Outlook Express.

And since then(1988) the POP3 is the active version.

Working of POP

All the incoming messages are stored on the POP server until the user login by using an email client and downloads the message to their computer. After the message is downloaded by the user it gets deleted from the server.

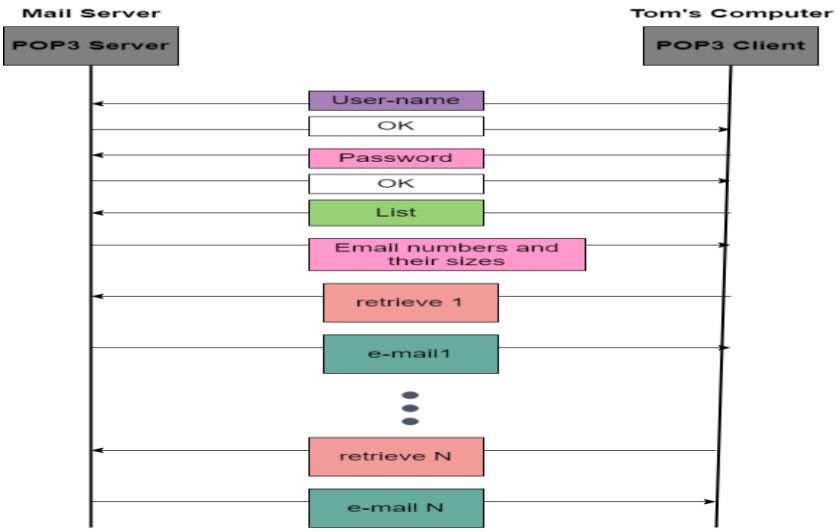
As we know that the SMTP is used to transfer the email message from the server to the server, basically POP is used to collect the email with an email client from the server and it does not include means to send messages.



If any user tries to check all the recent emails then they will establish a connection with the **POP3** at the server-side. The user sends the username and password to the server machine for getting the proper authentication. After getting the connection, users can receive all text-based emails and store them on their local terminal (machine), then finally discard all server copies and then breaks the connection from the server machine.

In order to retrieve a message from the server following steps are taken;

- Firstly a TCP connection is established by the client using port 110.
- The client identifies itself to the server.
- After that client issues a series of POP3 commands.



The above figure indicates the exchange of Commands and responses in the POP3

Features of POP protocol

Given below are some of the features provided by the POP protocol:

- The POP protocol uses PORT 110.
- It makes the use of a Persistent TCP connection.
- It is a Pull protocol.
- It is a connection-oriented protocol.
- The POP protocol is a stateful protocol until the mail is downloaded and across the sessions, it is a stateless protocol.

Let us now take a look at the commands of POP :

Commands	Description
LOGIN	This command is used to open a connection
STAT	This command is used to display the messages that are currently in the mailbox.
DELE	This command is used to delete a message.
RSET	This command is mainly used to reset the session to its initial state.
QUIT	This command is used to log off the session.
LIST	This command is mainly used to get the summary of each message where each message summary is shown.

Commands	Description
RETR	This command is mainly used to select a mailbox in order to access the messages.

Advantages of POP

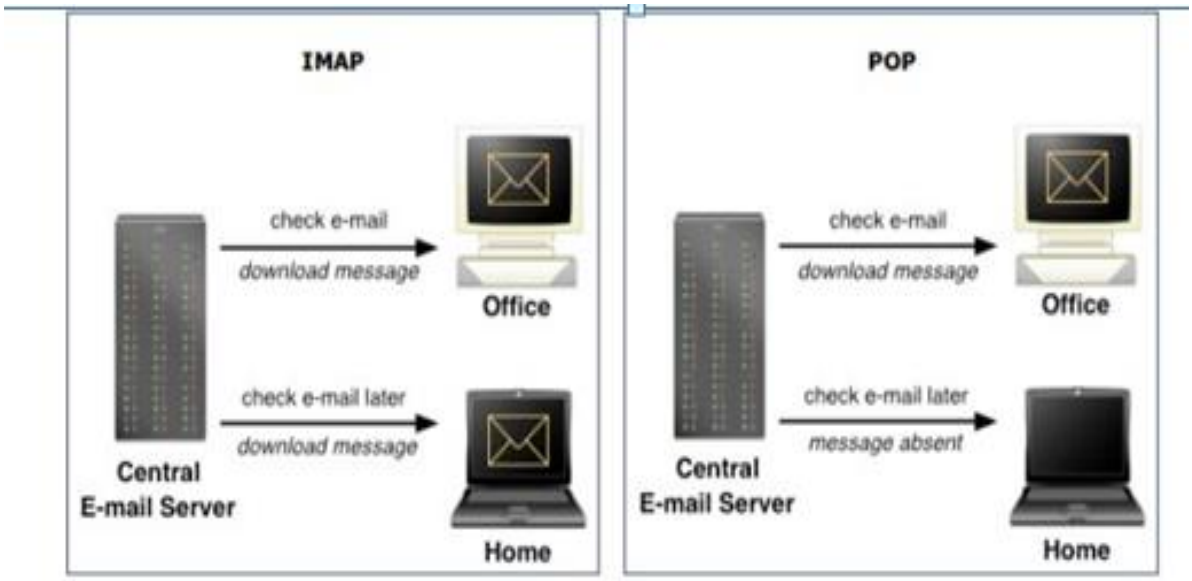
Given below are the advantages offered by the POP :

- This protocol does not require any internet connection in order to access the downloaded emails.
- In order to receive emails on a single device, POP3 is very useful.
- The Configuration of this protocol is simple and it is easy to use.
- Less storage space is needed in order to store emails on the hard disk.
- This protocol is much better for the ones who hardly check their email on any other computer.

Disadvantages of POP

Now it's time to take a look at the drawbacks of Post office Protocol(POP):

- The same email account cannot be accessed from multiple computers or devices.
- The spread of the virus is easily using this protocol because it is possible that the file attached with the email contains the virus.
- The transfer of the local email folder to another email client terminal point is a difficult task.



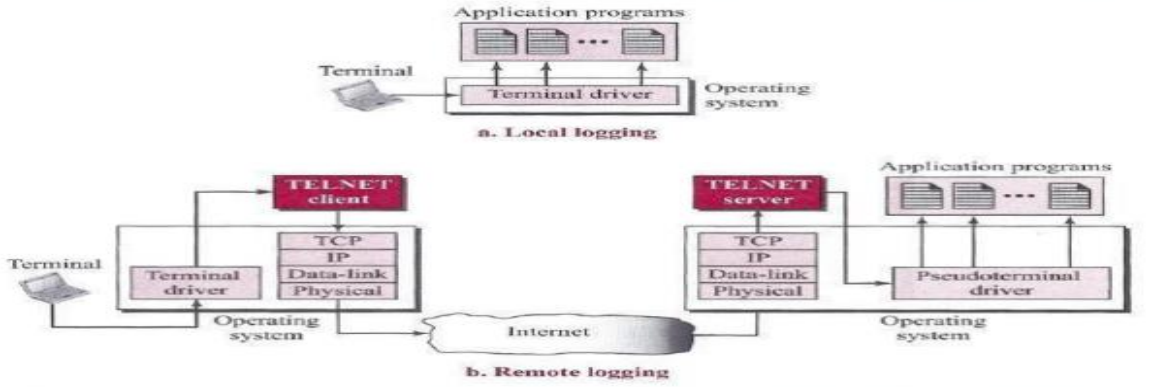
TELNET

One of the original remote logging protocols is **TELNET,** which is an abbreviation for *Terminal Network.*

A hacker can eavesdrop and obtain the logging name and password. Because of this security issue, the use of TELNET has diminished in favor of another protocol, Secure Shell (SSH),

Local versus Remote Logging

Local versus remote logging



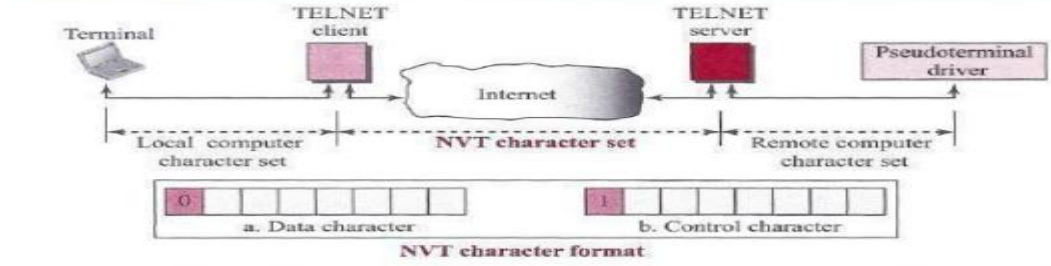
When a user logs into a local system, it is called *local logging.* As a user types at a terminal or at a workstation running a terminal emulator, the keystrokes are accepted by the terminal driver.

When a user wants to access an application program or utility located on a remote machine, she performs *remote logging.*

The characters are sent to the TELNET client, which transforms the characters into a universal character set called *Network Virtual Terminal* (NVT) characters

Network Virtual Terminal (NVT)

Concept of NVT



The mechanism to access a remote computer is complex. This is because every computer and its operating system accept a special combination of characters as tokens .We is dealing with heterogeneous systems. If we want to access any remote computer in the world TELNET solves this problem by defining a universal interface called the *Network Virtual Terminal (NVT)* character set.

The server TELNET, on the other hand, translates data and commands from NVT form into the form acceptable by the remote computer.

SECURE SHELL (SSH)

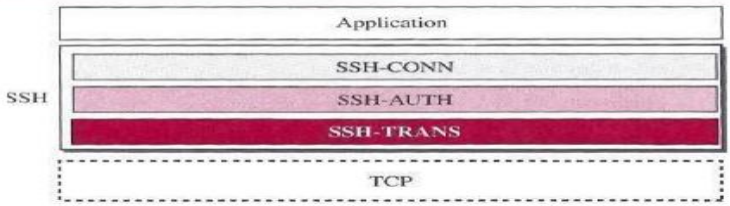
Secure Shell (SSH) is a secure application program that can be used today for several purposes such as remote logging and file transfer; it was originally designed to replace TELNET.

There are two versions of SSH: SSH-l and SSH-2

Components

SSH Transport-Layer Protocol (SSH-TRANS)

Components of SSH



SSH first uses a protocol that creates secured channel on top of the TCP.

When the procedure implementing this protocol is called, the client and server first use the TCP protocol to establish an insecure connection.

SSH Authentication Protocol (SSH-AUTH)

- After a secure channel is established between the client and the server and the server is authenticated for the client
- SSH can call another procedure that can authenticate the client for the server. The client authentication process in SSH is very similar to what is done in Secure Socket Layer (SSL)
- The request includes the user name, server name, the method of authentication, and the required data.

The server responds with either a success message, which confirms that the client is authenticated, or a failed message

SSH Connection Protocol (SSH-CONN)

One of the services provided by the SSH-CONN protocol is multiplexing. SSH-CONN takes the secure channel established by the two previous protocols and lets the client create multiple logical channels over it.

Each channel can be used for a different purpose, such as remote logging, file transfer, and so on

TELNET (Terminal Network):

- TELNET is client-server application that allows a user to log onto remote machine and lets the user to access any application program on a remote computer.
- TELNET uses the NVT (Network Virtual Terminal) system to encode characters on the local system.

- On the server (remote) machine, NVT decodes the characters to a form acceptable to the remote machine.
- TELNET is a protocol that provides a general, bi-directional, eight-bit byte oriented communications facility.
- Many application protocols are built upon the TELNET protocol □ Telnet services are used on PORT 23.

SSH protocol(Secure Shell)

The SSH protocol (also referred to as Secure Shell) is a method for secure remote login from one computer to another. It provides several alternative options for strong authentication, and it protects the communications security and integrity with strong encryption. It is a secure alternative to the non-protected login protocols (such as telnet, rlogin) and insecure file transfer methods (such as FTP).

TYPICAL USES OF THE SSH PROTOCOL

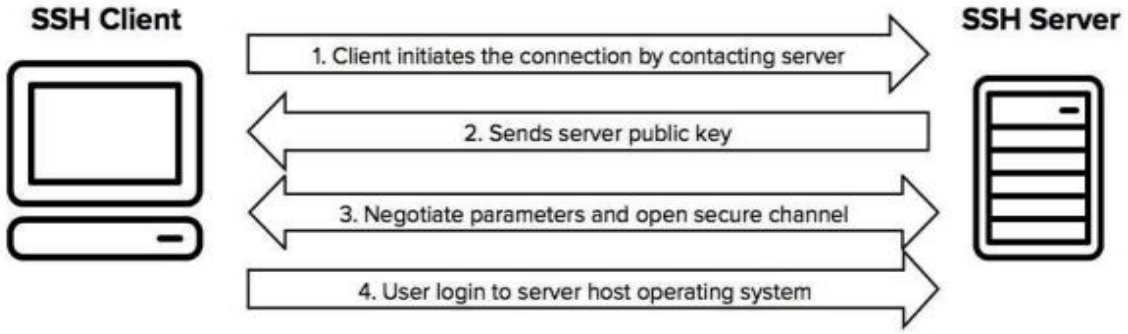
The protocol is used in corporate networks for:

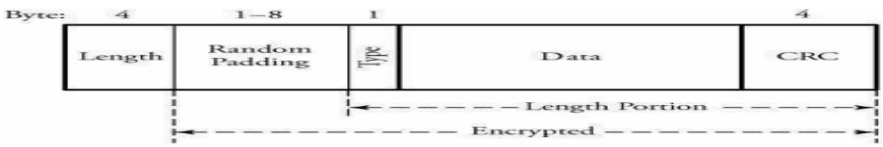
- providing secure access for users and automated processes
- interactive and automated file transfers
- issuing remote commands
- managing network infrastructure and other mission-critical system components.

HOW DOES THE SSH PROTOCOL WORK

The protocol works in the client-server model, which means that the connection is established by the SSH client connecting to the SSH server. The SSH client drives the connection setup process and uses public key cryptography to verify the identity of the SSH server. After the setup phase the SSH protocol uses strong symmetric encryption and hashing algorithms to ensure the privacy and integrity of the data that is exchanged between the client and server.

The figure below presents a simplified setup flow of a secure shell connection.





The format of an SSH packet

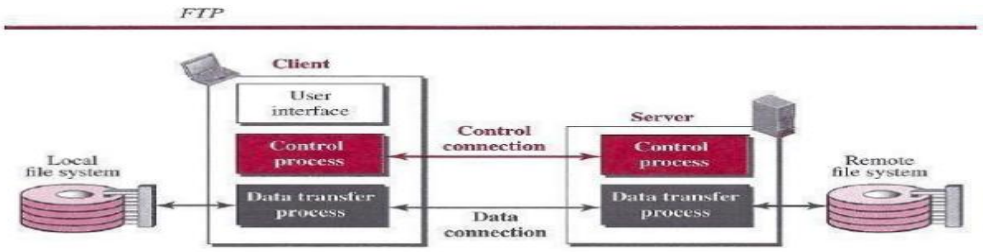
- Length indicates the size of the packet, not including the length field or the variablelength random padding field that follows it.
- Padding causes an intrusion to be more difficult. o Type identifies the type of message. o CRC , or cyclic redundancy check, is an error-detection field .

When encryption is enabled, all fields except length are encrypted. SSH also permits optional compression of the data, which is useful when SSH is used in low-bandwidth situations. In such cases, the client and the server negotiate compression, and only the type and data fields are compressed.

FTP

File Transfer Protocol (FTP) is the standard protocol provided by *TCP/IP* for copying a file from one host to another.

Two systems may have different directory structures. All of these problems have been solved by FTP in a very simple and elegant approach.



Two Connections

- The control connection remains connected during the entire interactive FTP session.
- 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.
- FTP uses two well-known TCP ports: port 21 is used for the control connection, and port 20 is used for the data connection.

File Transfer Protocol

- The File Transfer Protocol (FTP) is the most widely used protocol for file transfer over the network. FTP uses TCP/IP for communication and it works on TCP port 21. FTP works on Client/Server Model where a client requests file from Server and server sends requested resource back to the client.
- FTP uses out-of-band controlling i.e. FTP uses TCP port 20 for exchanging controlling information and the actual data is sent over TCP port 21.
- The client requests the server for a file. When the server receives a request for a file, it opens a TCP connection for the client and transfers the file. After the transfer is complete, the server closes the connection. For a second file, client requests again and the server reopens a new TCP connection.
- FTP is the standard mechanism provided by TCP/IP for copying a file from one host to another.
- FTP differs form other client-server applications because it establishes 2 connections between hosts.
- Two connections are: Data Connection and Control Connection.
- Data Connection uses PORT 20 for the purpose and control connection uses PORT 21 for the purpose.

- FTP is built on a client-server architecture and uses separate control and data connections between the client and the server.
- One connection is used for data transfer, the other for control information (commands and responses).
- It transfer data reliably and efficiently.

Secure Copy Protocol(SCP)

Secure Copy, or SCP, does not use FTP or SSL to transfer files, rather Secure Copy handles the file transfer and relies on the SSH protocol to provide authentication and security for both credentials and data. Unfortunately, SCP doesn't have file management capabilities -- certainly a cause of concern. When an SCP client sends a request to download files or directories, the server feeds the client with its subdirectories and files, causing a server-driven download. This makes the protocol a security risk if the server is malicious or has been compromised. You will find that SCP is being replaced by the more comprehensive and platform-independent SFTP protocol, which is also based on SSH.

TFTP:

TFTP stands for Trivial File Transfer Protocol. TFTP is used to transfer a file either from client to server or from server to client without the need of FTP feature. Software of TFTP is smaller than FTP. TFTP works on 69 Port number and its service is provided by UDP.

Now, we shall see the difference between FTP and TFTP:

S.NO	FTP	TFTP
1.	FTP stands for File Transfer Protocol.	TFTP stands for Trivial File Transfer Protocol.
2.	The software of FTP is larger than TFTP.	While software of TFTP is smaller than FTP.
3.	FTP works on two ports: 20 and 21.	While TFTP works on 69 Port number.
4.	FTP services are provided by TCP.	While TFTP services are provided by UDP.
5.	The complexity of FTP is higher than TFTP.	While the complexity of TFTP is less than FTP complexity.
6.	There are many commands or messages in FTP.	There are only 5 messages in TFTP.
7.	FTP need authentication for communication.	While TFTP does not need authentication for

	FTP is generally suited for uploading and	communication.
8.	downloading of files by remote users.	While TFTP is mainly used for transmission of configurations to and from network devices.
9.	FTP is a reliable transfer protocol.	While; TFTP is an unreliable transfer protocol.
10.	FTP is based on TCP.	While; TFTP is based on UDP.
11.	FTP is slower.	TFTP is faster as compared to FTP.

BOOTP

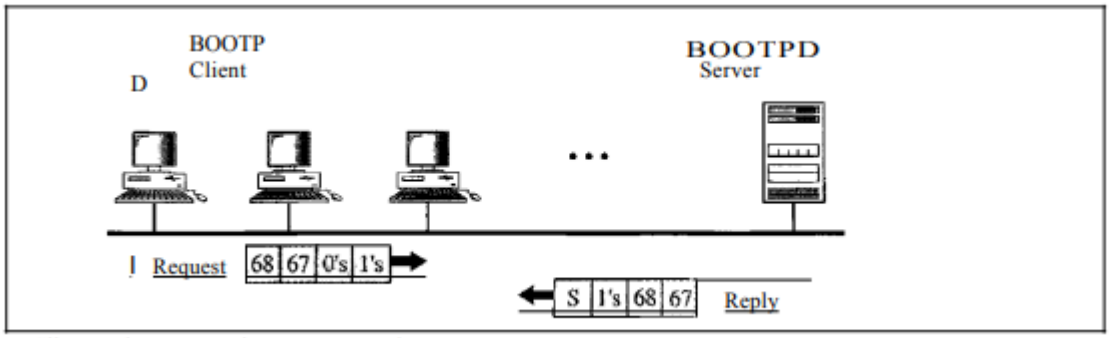
BOOTP(Bootstrap Protocol) is a client/server protocol designed to provide physical address to logical address mapping

BOOTP is an application layer protocol

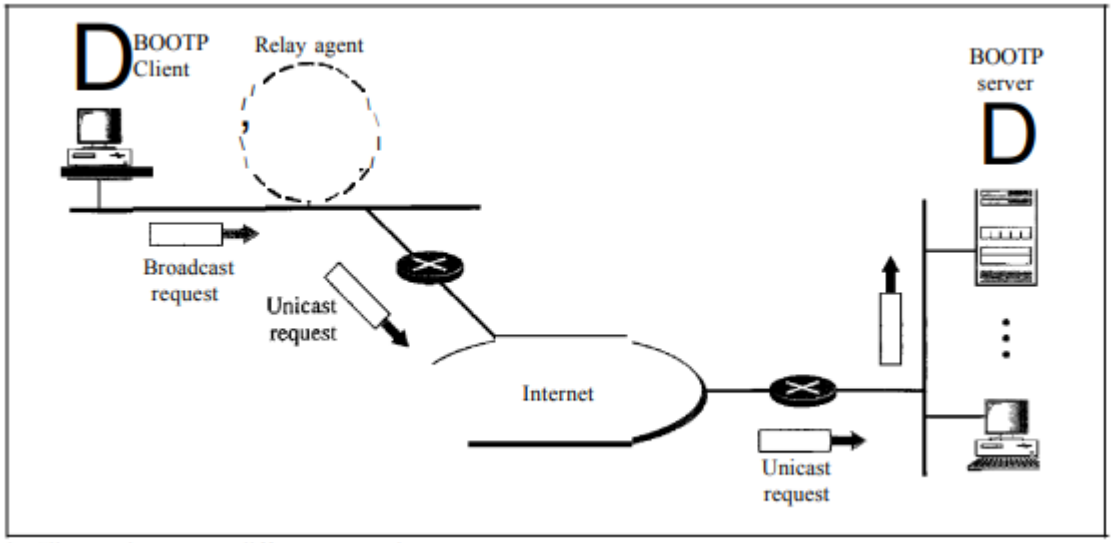
BOOTP is not a dynamic configuration protocol

DHCP(Dynamic Host Configuration Protocol) provides static and dynamic address allocation that can be manual or automatic

BOOTP client and server on the same and different network



a. Client and server on the same network



b. Client and server on different networks

The reader may ask how a client can send an IP datagram when it knows neither its own IP address (the source address) nor the server's IP address (the destination address). The client simply uses all as as the source address and a11Is as the destination address.

One of the advantages of BOOTP over RARP is that the client and server are application-layer processes. As in other application-layer processes, a client can be in one network and the server in another, separated by several other networks. However, there is one problem that must be solved. The BOOTP request is broadcast because the client does not know the IP address of the server. A broadcast IP datagram cannot pass through any router. To solve the problem, there is a need for an intermediary. One of the hosts (or a router that can be configured to operate at the application layer) can be used as a relay. The host in this case is called a relay agent. The relay agent knows the unicast address of a BOOTP server. When it receives this type of packet, it encapsulates the message in a unicast datagram and sends the request to the BOOTP server. The packet,

It is an application layer protocol hence it can be implemented in any application.

There is no need to assign an IP address to each machine. Only when a machine needs an IP address, it can be assigned. Thus, the network can operate with only a limited set of IP addresses.

A common BOOTP server can be used for multiple networks which are managed by a single administrator.

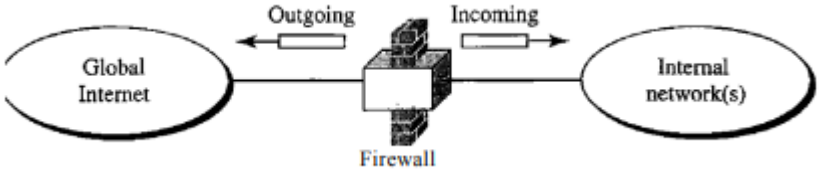
Limitations of BOOTP

i. BOOTP requires a predefined mapping between physical and logical address in the server. Hence, it is not a dynamic configuration protocol.
ii. The table has to be manually updated.
iii. The communication between client and server is done only when the system starts.
iv. It does not support dynamic addressing.
v. Unused IP addresses cannot be automatically obtained by a client.
vi. Manually and statically assigning IP addresses to hosts requires accurate and up-to-date documentation in order to prevent duplicate IP address problems.

Firewall

A firewall is a network security device, either hardware or software-based, which monitors all incoming and outgoing traffic and based on a defined set of security rules it accepts, rejects or drops that specific traffic.

Accept : allow the traffic
Reject : block the traffic but reply with an "unreachable error"
Drop : block the traffic with no reply
A firewall establishes a barrier between secured internal networks and outside untrusted network, such as the Internet.

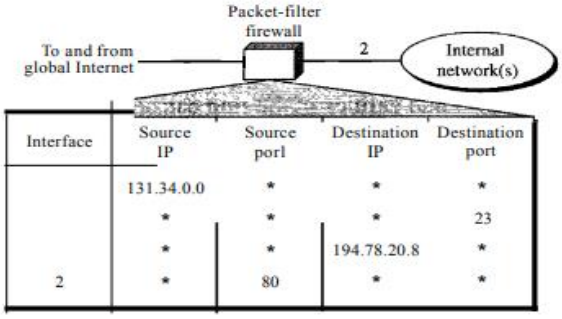


For example, a firewall may filter all incoming packets destined for a specific host or a specific server such as HTTP. A firewall can be used to deny access to a specific host or a specific service in the organization.

A firewall is usually classified as a packet-filter firewall or a proxy-based firewall.

Packet-Filter Firewall

A firewall can be used as a packet filter. It can forward or block packets based on the information in the network layer and transport layer headers: source and destination IP addresses, source and destination port addresses, and type of protocol (TCP or UDP). A packet-filter firewall is a router that uses a filtering table to decide which packets must be discarded (not forwarded). Figure 32.23 shows an example of a filtering table for this kind of a firewall.



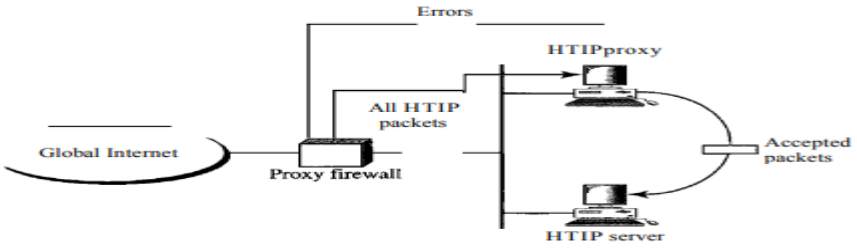
Interface	Source IP	Source porl	Destination IP	Destination port
	131.34.0.0	*	*	*
	*	*	*	23
	*	*	194.78.20.8	*
2	*	80	*	*

). Incoming packets from network 131.34.0.0 are blocked (security precaution). Note that the * (asterisk) means "any."
2. Incoming packets destined for any internal TELNET server (port 23) are blocked.
3. Incoming packets destined for internal host 194.78.20.8 are blocked. The organization wants this host for internal use only.
4. Outgoing packets destined for an HTfP server (port 80) are blocked. The organization does not want employees to browse the Internet.

Proxy Firewall

The packet-filter firewall is based on the information available in the network layer and transport layer headers (IP and TCPIUDP). However, sometimes we need to filter a message based on the information available in the message itself (at the application layer). As an example, assume that an organization wants to implement the following policies regarding its Web pages: Only those Internet users who have previously established business relations with the company can have access; access to other users must be blocked. In this case, a packet-filter firewall is not feasible because it cannot distinguish between different packets arriving at TCP port 80 (HTTP). Testing must be done at the application level (using URLs).

One solution is to install a proxy computer (sometimes called an application gateway), which stands between the customer (user client) computer and the corporation



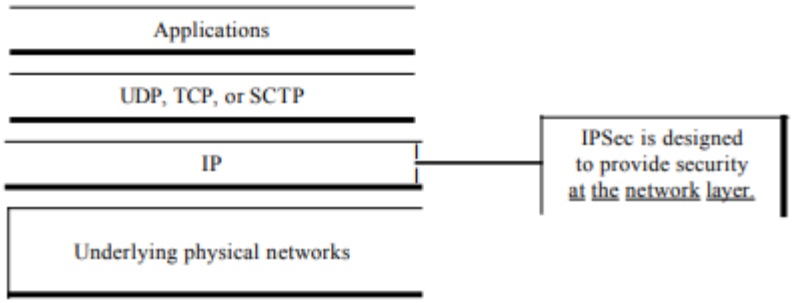
When the user client process sends a message, the proxy firewall runs a server process to receive the request. The server opens the packet at the application level and finds out if the request is legitimate. If it is, the server acts as a client process and sends the message to the real server in the corporation. If it is not, the message is dropped and an error message is sent to the external user. In this way, the requests of the external users are filtered based on the contents at the application layer. Figure 32.24 shows a proxy firewall implementation.

A proxy firewall filters at the application layer.

IPSecurity (IPSec)

IPSecurity (IPSec) is a collection of protocols designed by the Internet Engineering Task Force (IETF) to provide security for a packet at the network level.

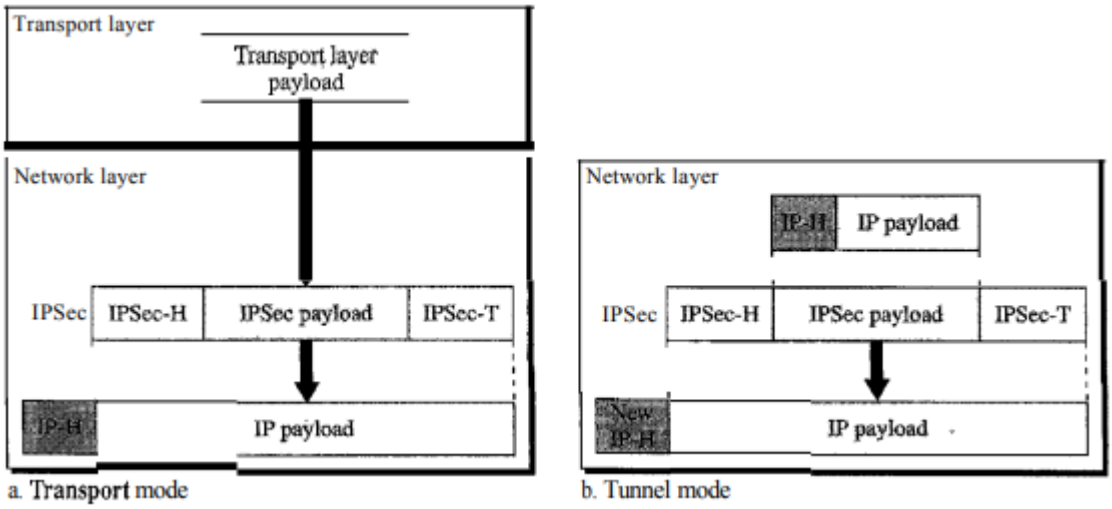
TCP/IP protocol suite and IPSec



Two Modes

IPSec operates in one of two different modes: the transport mode or the tunnel mode as

Transport mode and tunnel modes of IPSec protocol

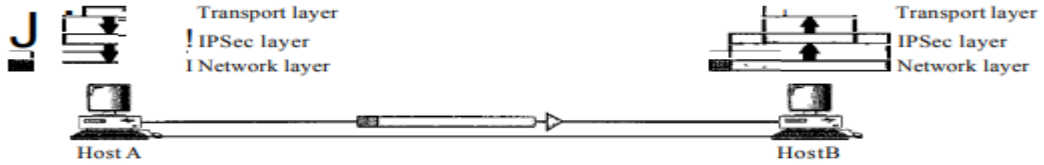


Transport Mode

In the transport mode, IPSec protects what is delivered from the transport layer to the network layer. In other words, the transport mode protects the network layer payload, the payload to be encapsulated in the network layer.

IPSec in the transport mode does not protect the IP header; it only protects the information coming from the transport layer.

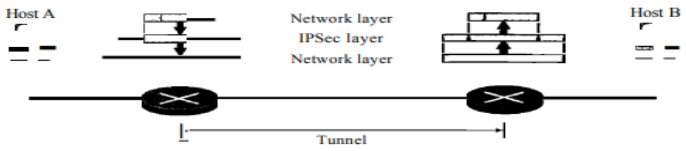
Transport mode in action



Tunnel Mode

In the tunnel mode, IPSec protects the entire IP packet. It takes an IP packet, including the header, applies IPSec security methods to the entire packet, and then adds a new IP

Tunnel mode in action



IPSec in tunnel mode protects the original IP header.

Two Security Protocols

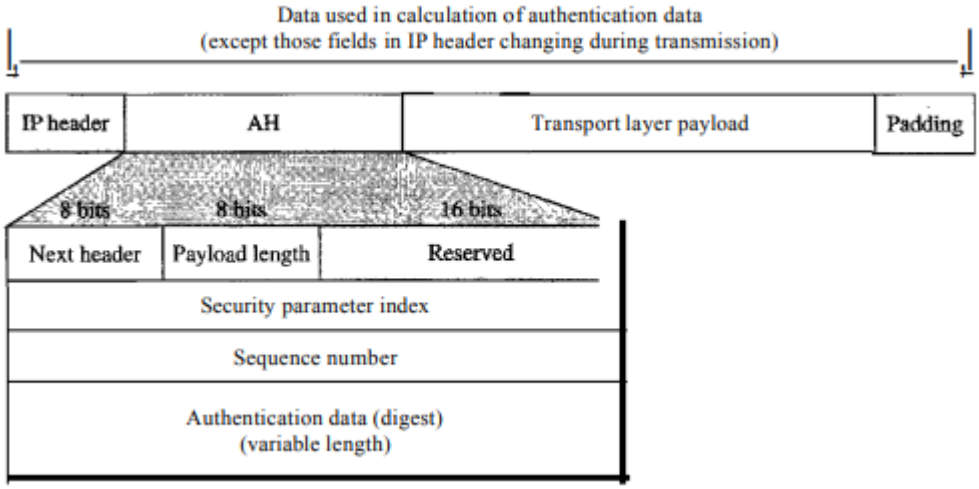
IPSec defines two protocols-the Authentication Header (AH) Protocol and the Encapsulating Security Payload (ESP) Protocol-to provide authentication and/or encryption for packets at the IP level.

Authentication Header (AH)

The Authentication Header (AH) Protocol is designed to authenticate the source host and to ensure the integrity of the payload carried in the IP packet. The protocol uses a hash function and a symmetric key to create a message digest; the digest is inserted in

1. An authentication header is added to the payload with the authentication data field set to zero.

Authentication Header (AH) Protocol in transport mode



2. Padding may be added to make the total length even for a particular hashing algorithm.
3. Hashing is based on the total packet. However, only those fields of the IP header that do not change during transmission are included in the calculation of the message digest (authentication data).
4. The authentication data are inserted in the authentication header.
5. The IP header is added after the value of the protocol field is changed to 51.

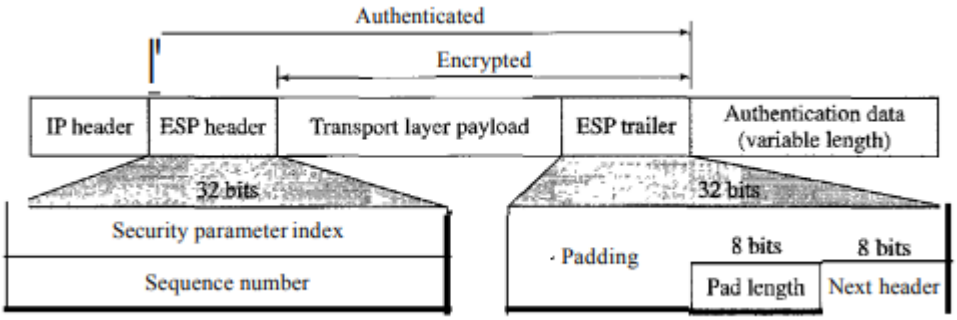
A brief description of each field follows:

- Next header. The 8-bit next-header field defines the type of payload carried by the IP datagram (such as TCP, UDP, ICMP, or OSPF). It has the same function as the protocol field in the IP header before encapsulation. In other words, the process copies the value of the protocol field in the IP datagram to this field. The value of the protocol field in the new IP datagram is now set to 51 to show that the packet carries an authentication header.
- Payload length. The name of this 8-bit field is misleading. It does not define the length of the payload; it defines the length of the authentication header in 4-byte multiples, but it does not include the first 8 bytes.
- Security parameter index. The 32-bit security parameter index (SPI) field plays the role of a virtual-circuit identifier and is the same for all packets sent during a connection called a security association (discussed later).
- Sequence number. A 32-bit sequence number provides ordering information for a sequence of datagrams. The sequence numbers prevent a playback. Note that the sequence number is not repeated even if a packet is retransmitted. A sequence number does not wrap around after it reaches 2^{32}; a new connection must be established.
- Authentication data. Finally, the authentication data field is the result of applying a hash function to the entire IP datagram except for the fields that are changed during transit (e.g., time-to-live),

The AH Protocol provides source authentication and data integrity, but not privacy.

Encapsulating Security Payload (ESP)

The AH Protocol does not provide privacy, only source authentication and data integrity. IPSec later defined an alternative protocol that provides source authentication, integrity, and privacy called Encapsulating Security Payload (ESP). ESP adds a header and



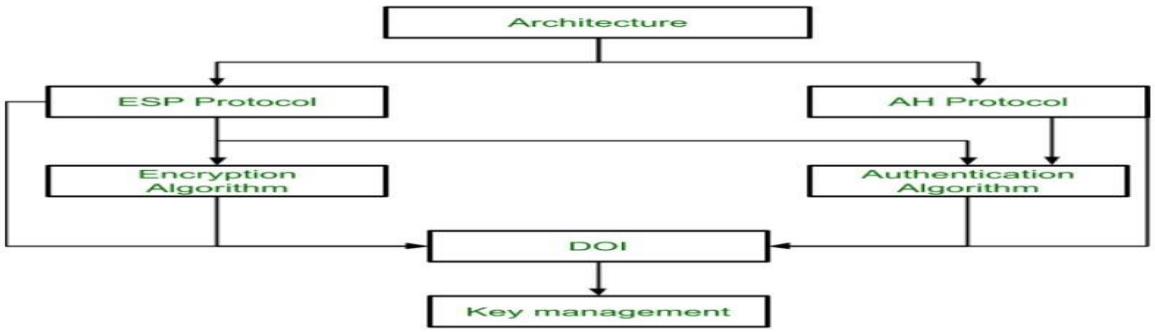
When an IP datagram carries an ESP header and trailer, the value of the protocol field in the IP header is 50. A field inside the ESP trailer (the next-header field) holds the original value of the protocol field (the type of payload being carried by the IP datagram, such as TCP or UDP). The ESP procedure follows these steps:

1. An ESP trailer is added to the payload.
2. The payload and the trailer are encrypted.
3. The ESP header is added.
4. The ESP header, payload, and ESP trailer are used to create the authentication data.
5. The authentication data are added to the end of the ESP trailer.
6. The IP header is added after the protocol value is changed to 50.

The fields for the header and trailer are as follows:

- Security parameter index. The 32-bit security parameter index field is similar to that defined for the AH Protocol.
- Sequence number. The 32-bit sequence number field is similar to that defined for the AH Protocol.
- Padding. This variable-length field (0 to 255 bytes) of 0s serves as padding.
- Pad length. The 8-bit pad length field defines the number of padding bytes. The value is between 0 and 255; the maximum value is rare.
- Next header. The 8-bitnext-header field is similar to that defined in theAH Protocol. It serves the same purpose as the protocol field in the IP header before encapsulation.

IP Security Architecture:

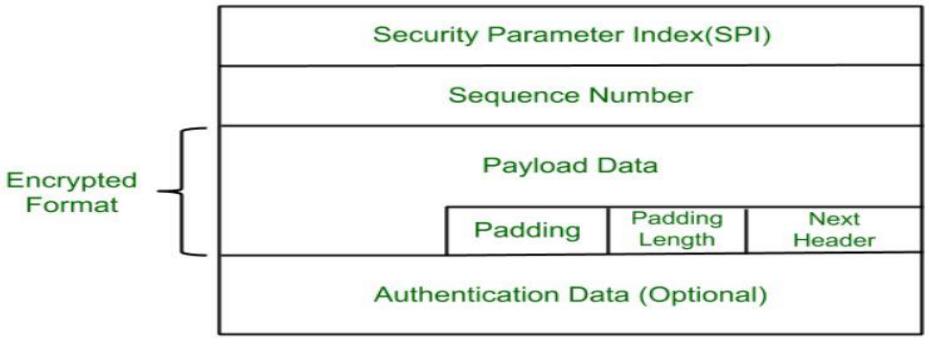


1. Architecture:
Architecture or IP Security Architecture covers the general concepts, definitions, protocols, algorithms and security requirements of IP Security technology.

2. ESP Protocol:
ESP(Encapsulation Security Payload) provide the confidentiality service. Encapsulation Security Payload is implemented in either two ways:

- ESP with optional Authentication.
- ESP with Authentication.

Packet Format:



- **Security Parameter Index(SPI):**
  This parameter is used in Security Association. It is used to give a unique number to the connection build between Client and Server.

- **Sequence Number:**
  Unique Sequence number are allotted to every packet so that at the receiver side packets can be arranged properly.

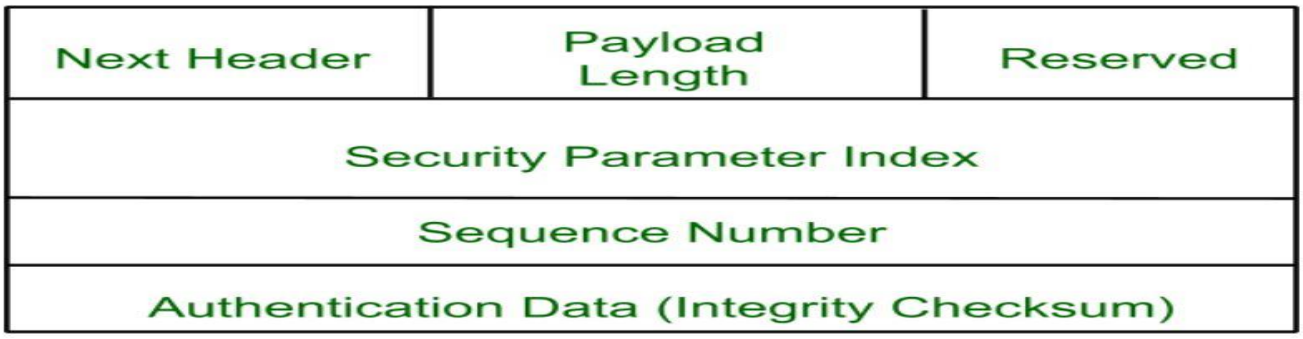
- **Payload Data:**
  Payload data means the actual data or the actual message. The Payload data is in encrypted format to achieve confidentiality.

- **Padding:**
  Extra bits or space added to the original message in order to ensure confidentiality. Padding length is the size of the added bits or space in the original message.

- **Next Header:**
  Next header means the next payload or next actual data.

- **Authentication Data**
  This field is optional in ESP protocol packet format.

3. Encryption algorithm:
Encryption algorithm is the document that describes various encryption algorithm used for Encapsulation Security Payload.
4. AH Protocol:
AH (Authentication Header) Protocol provides both Authentication and Integrity service. Authentication Header is implemented in one way only: Authentication along with Integrity.



Authentication Header covers the packet format and general issue related to the use of AH for packet authentication and integrity.

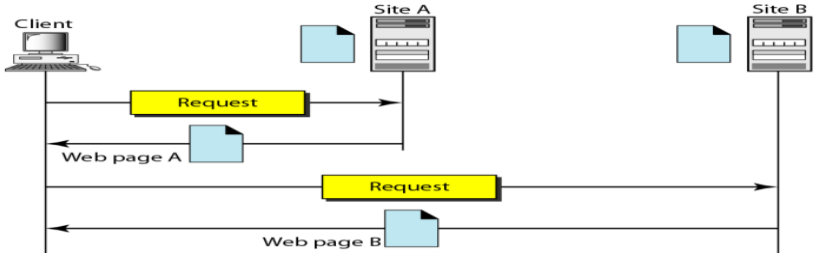
5. Authentication Algorithm:
Authentication Algorithm contains the set of the documents that describe authentication algorithm used for AH and for the authentication option of ESP.
6. DOI (Domain of Interpretation):
DOI is the identifier which support both AH and ESP protocols. It contains values needed for documentation related to each other.
7. Key Management:
Key Management contains the document that describes how the keys are exchanged between sender and receiver

WWW

The WWW today is a distributed client/server service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called sites as shown in fig. Client (Browser) Server Uniform Resource Locator Cookies

Topics discussed in this section:
Client (Browser)
Server
Uniform Resource Locator
Cookies

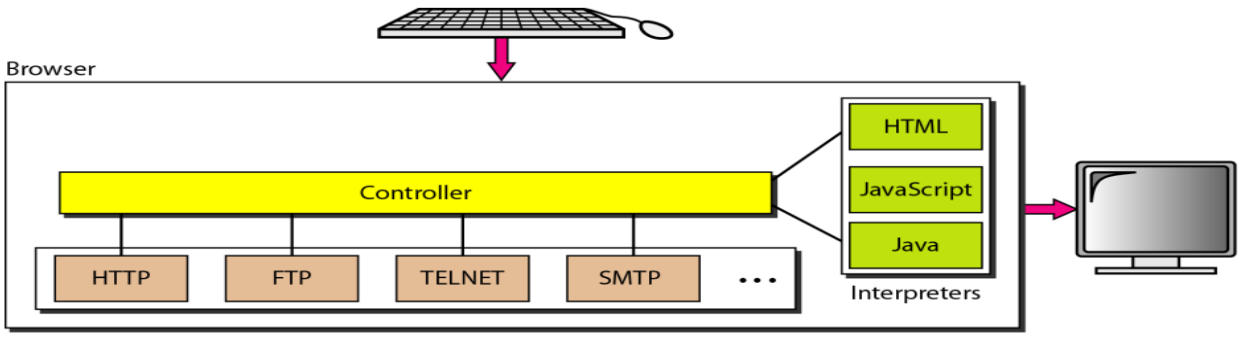
Architecture of WWW



Client (Browser) A variety of vendors offer commercial browsers that interpret and display a Web document,and all use nearly the same architecture. Each browser usually consists of three parts: a controller, client protocol, and interpreters. The controller receives input from the keyboard or the mouse and uses the client programs to access the document. After the document has been accessed, the controller uses one of the interpreters to display the document on the screen. The interpreter can be HTML, Java, or JavaScript, depending on the type of document The client protocol can be one of the protocols described previously such as FTP or HTTP.

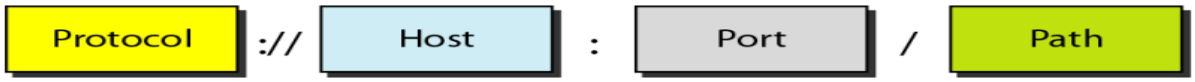
Server The Web page is stored at the server. Each time a client request arrives, the corresponding document is sent to the client. To improve efficiency, servers normally store requested files in a cache in memory; memory is faster to access than disk. A server can also become more efficient through multithreading or multiprocessing. In this case, a server can answer more than one request at a time. 2

Browser



Uniform Resource Locator A client that wants to access a Web page needs the address. To facilitate the access of documents distributed throughout the world, HTTP uses locators. The uniform resource locator (URL) is a standard for specifying any kind of information on the Internet. The URL defines four things: protocol, host computer, port, and path. The protocol is the client/server program used to retrieve the document. Many different protocols can retrieve a document; among them are FTP or HTTP. The most common today is HTTP. The host is the computer on which the information is located, although the name of the computer can be an alias. Web pages are usually stored in computers, and computers are given alias names that usually begin with the characters "www". The URL can optionally contain the port number of the server. If the port is included, it is inserted between the host and the path, and it is separated from the host by a colon. Path is the pathname of the file where the information is located. Note that the path can itself contain slashes that, in the UNIX operating system, separate the directories from the subdirectories and files.

URL



An HTTP cookie (also called web cookie, Internetcookie, browser cookie or simply cookie, the latter which is not to be confused with the literal definition), is a small piece of data sent from a website and stored in a user's web browser while the user is browsing that website

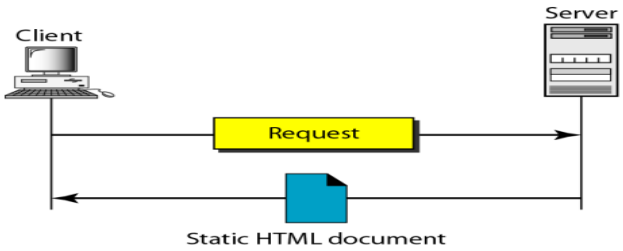
WEB DOCUMENTS

The documents in the WWW can be grouped into three broad categories: static, dynamic, and active. The category is based on the time at which the contents of the document are determined.

Static Documents Static documents are fixed-content documents that are created and stored in a server. The client can get only a copy of the document. When a client accesses the document, a copy of the document is sent. The user can ther use a browsing program to display the document

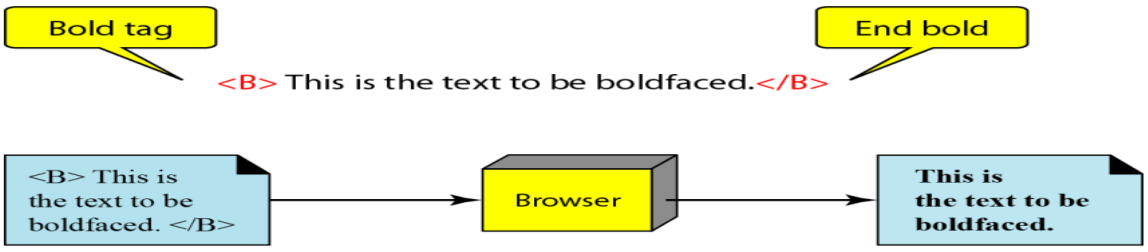
Topics discussed in this section:

Static Documents
Dynamic Documents
Active Documents

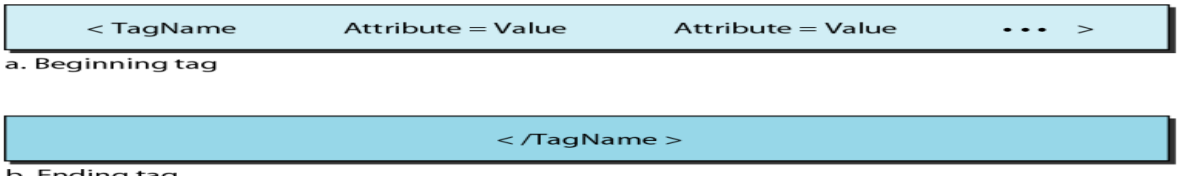


Boldface tags

HTML
Hypertext Markup Language (HTML) is a language for creating Web pages.



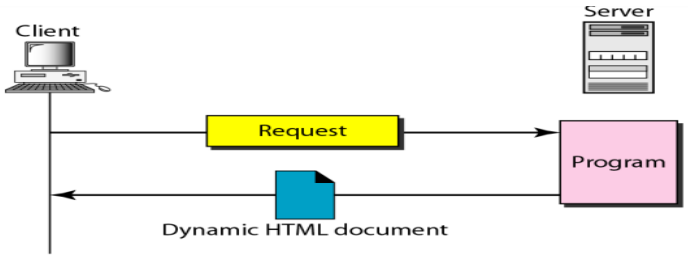
Beginning and ending tags



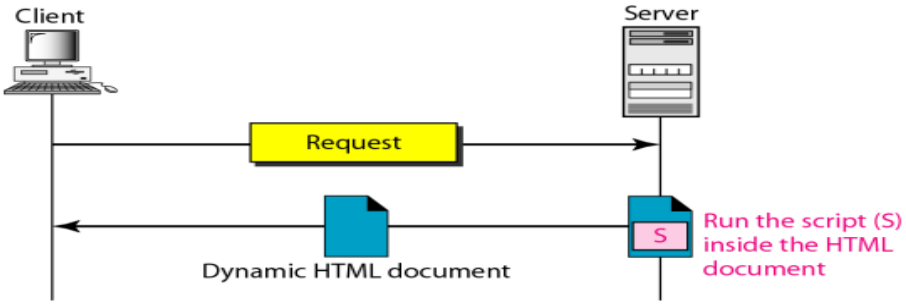
Dynamic Documents A dynamic document is created by a Web server whenever a browser requests the document. When a request arrives, the Web server runs an application program or a script that creates the dynamic document. The server returns the output of the program or script as a response to the browser that requested the document. A very simple

example of a dynamic document is the retrieval of the time and date from a server. Time and date are kinds of information that are dynamic in that they change from moment to moment. The client can ask the server to run a program such as the date program in UNIX and send the result of the program to the client. Common Gateway Interface (CGI) The Common Gateway Interface (CGI) is a technology that creates and handles dynamic documents. Hypertext Preprocessor (pHP), which uses the Perl language; Java Server Pages (JSP), which uses the Java language for scripting; Active Server Pages (ASP), a Microsoft product which uses Visual Basic language for scripting; and ColdFusion, which embeds SQL database queries in the HTML document

Dynamic document using CGI



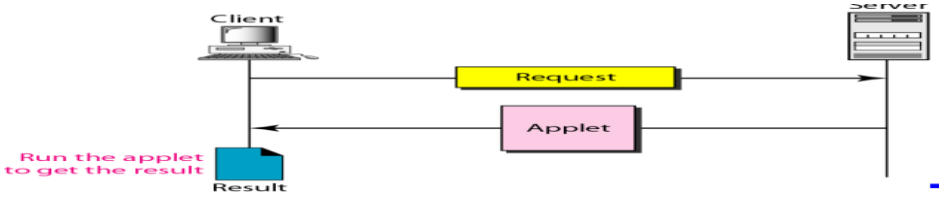
Dynamic document using server-site script



Dynamic documents are sometimes referred to as server-site dynamic documents.

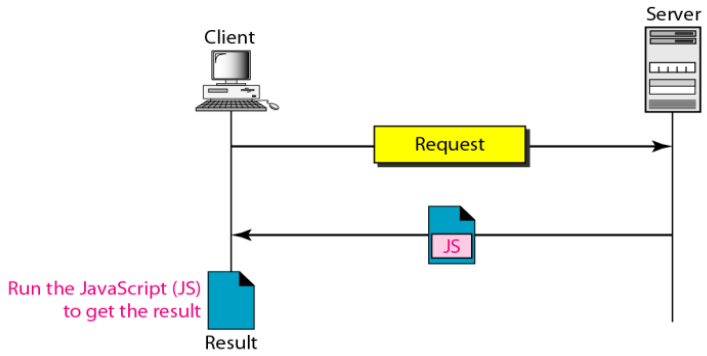
Active document using Java applet

Active Documents
For many applications, we need a program or a script to be run at the client site. These are called active documents



37

Active document using client-site script

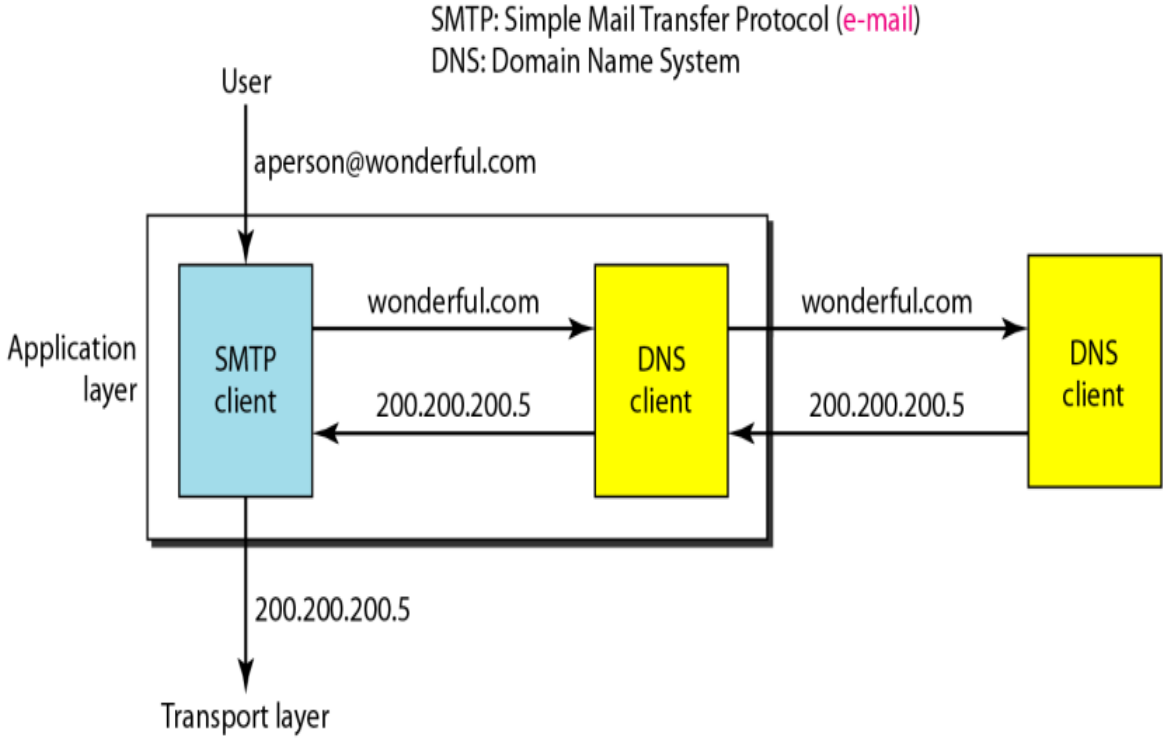


Active documents are sometimes referred to as client-site dynamic documents.

HTTP version 1.1 specifies a persistent connection by default.

DNS (Domain Name System)

To identify an entity, TCP/IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet. However, people prefer to use names instead of numeric addresses. Therefore, we need a system that can map a name to an address or an address to a name.



NAME SPACE

A name space that maps each address to a unique name can be organized in two ways: fiat or hierarchical.

Flat Name Space

In a flat name space, a name is assigned to an address. A name in this space is a sequence of characters without structure.

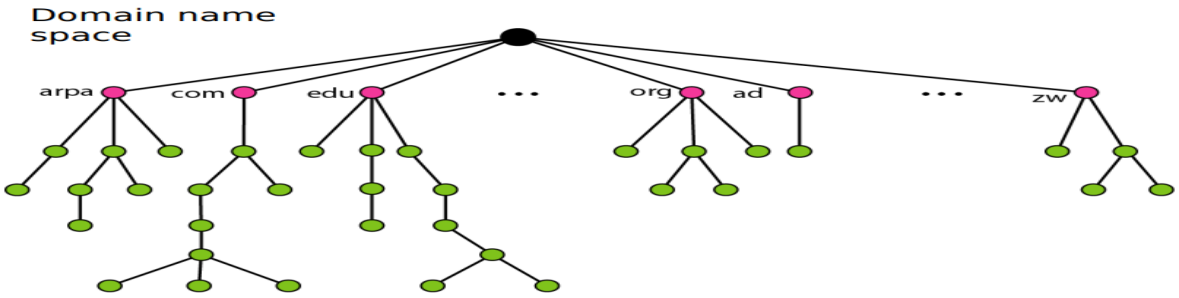
Hierarchical Name Space

In a hierarchical name space, each name is made of several parts. The first part can define the nature of the organization, the second part can define the name of an organization, the third part can define departments in the organization, and so on.

Exa: *challenger.jhda.edu, challenger.berkeley.edu, and challenger.smart.com*

DOMAIN NAME SPACE

To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127.



Label

Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string (empty string). DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

Domain Name

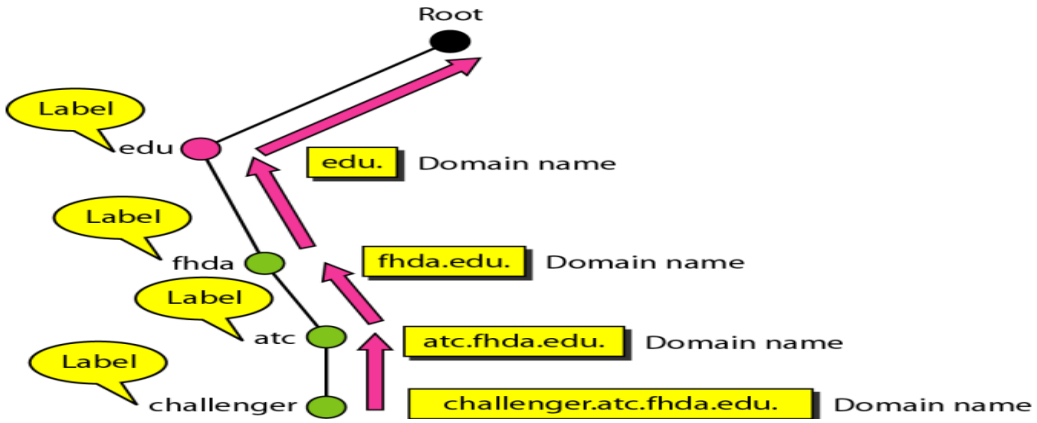
Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.). The domain names are always read from the node up to the root. The last label is the label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing. Below Figure shows some domain names

Figure shows some domain names

-

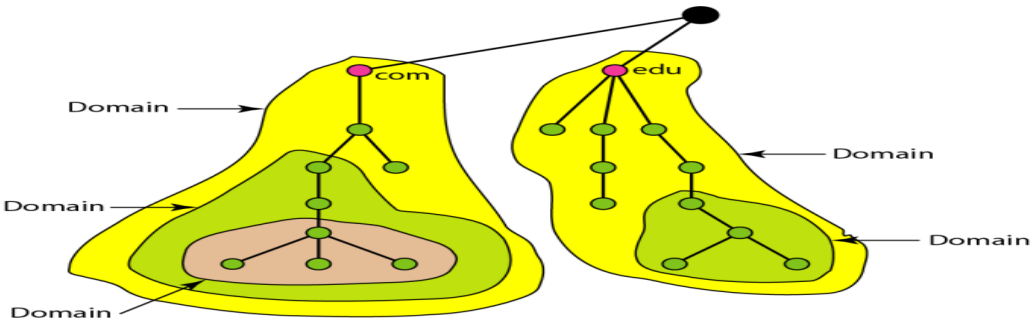
FQDN	PQDN
challenger.atc.fhda.edu. cs.hmme.com. www.funny.int.	challenger.atc.fhda.edu cs.hmme www

Domain names and labels



Domain

A domain is a subtree of the domain name space. The name of the domain is the domain name of the node at the top of the subtree.

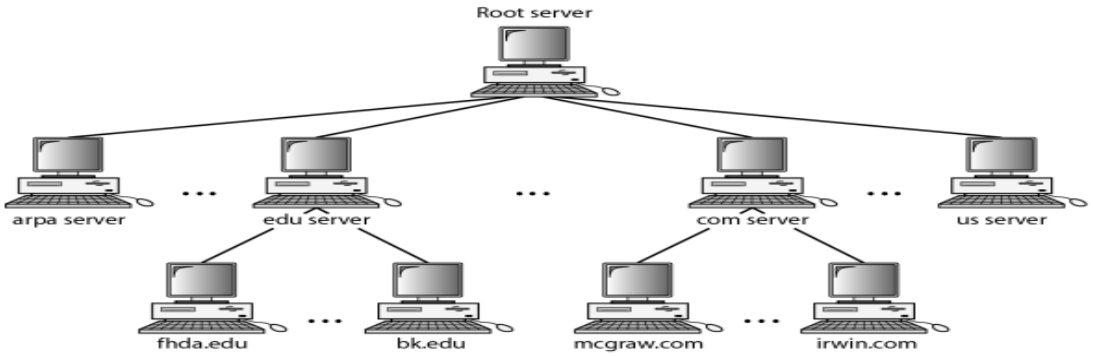


DISTRIBUTION OF NAME SPACE:

The information contained in the domain name space must be stored. However, it is very inefficient and also unreliable to have just one computer store such a huge amount of information. In this section, we discuss the distribution of the domain name space

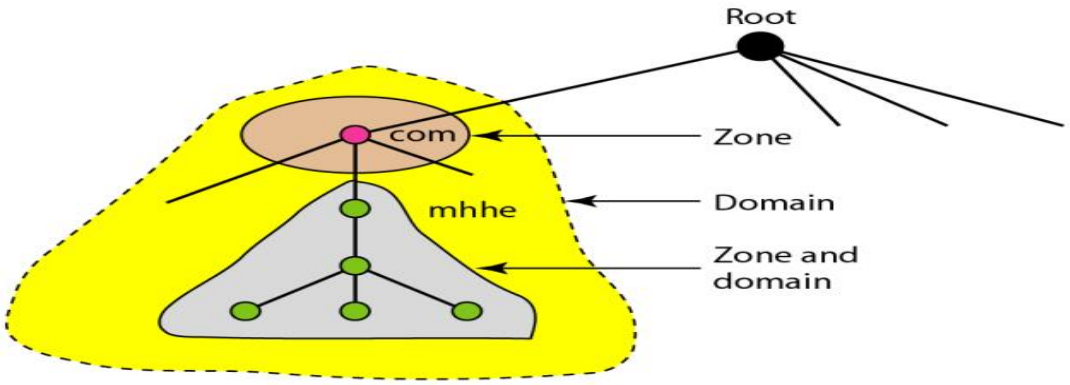
1 Hierarchy of Name Servers

distribute the information among many computers called DNS servers. we let the root stand alone and create as many domains (subtrees) as there are first-level nodes



2 Zone

Since the complete domain name hierarchy cannot be stored on a single server, it is divided among many servers. What a server is responsible for or has authority over is called a zone. We can define a zone as a contiguous part of the entire tree



3 Root Server

A root server is a server whose zone consists of the whole tree. A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers. There are several root servers, each covering the whole domain name space. The servers are distributed all around the world.

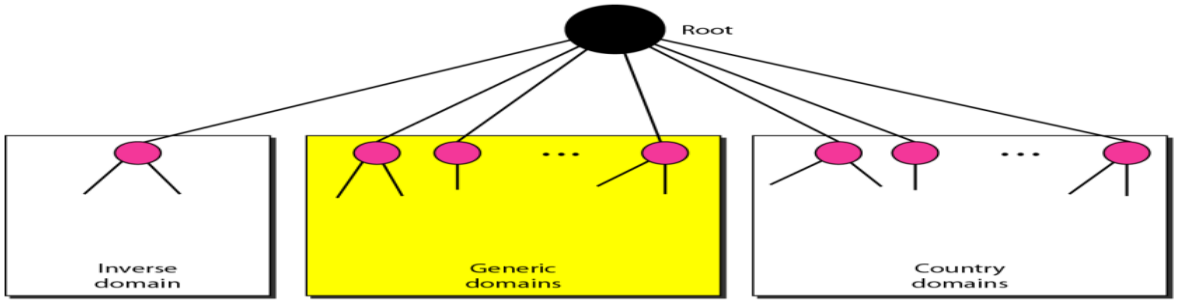
4 Primary and Secondary Servers

A primary server is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk

A secondary server is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files

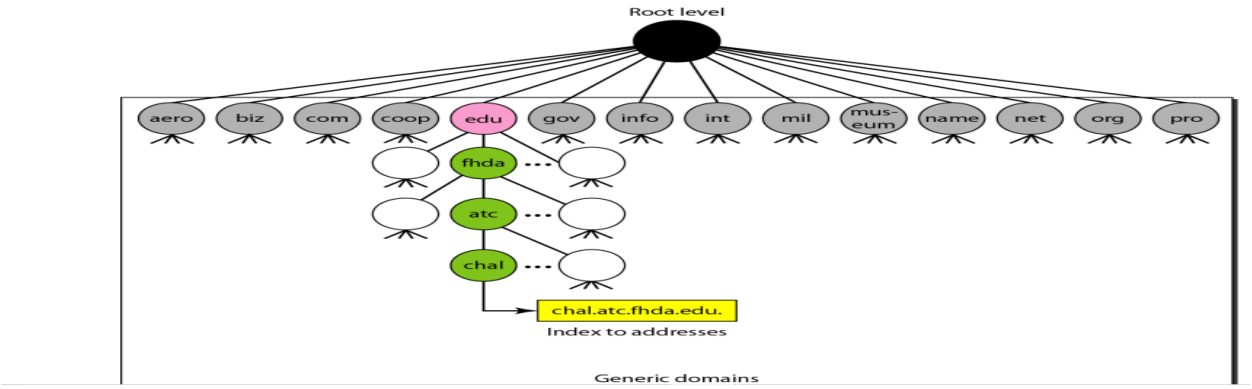
DNS IN THE INTERNET

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



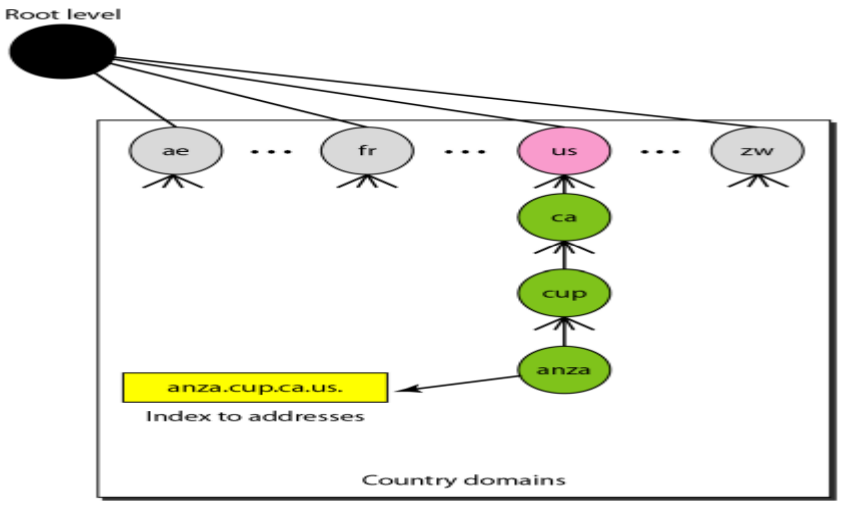
1 Generic Domains

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



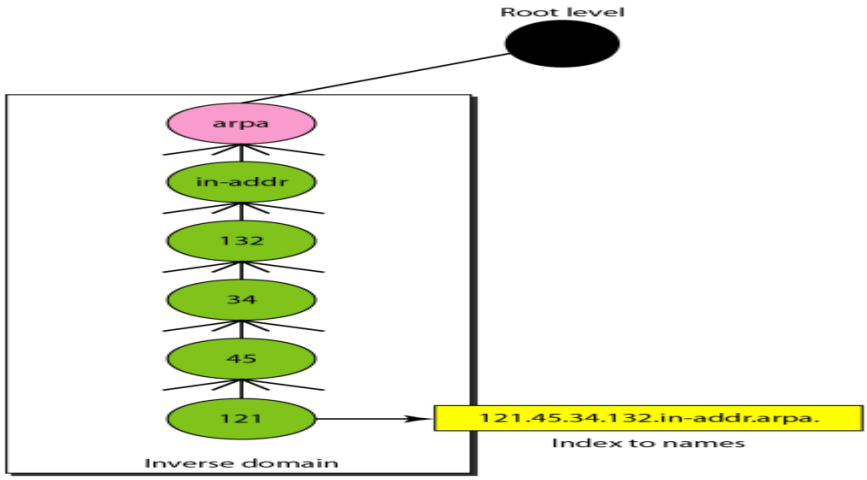
2 Country Domains

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.).



3 Inverse Domain

The inverse domain is used to map an address to a name.



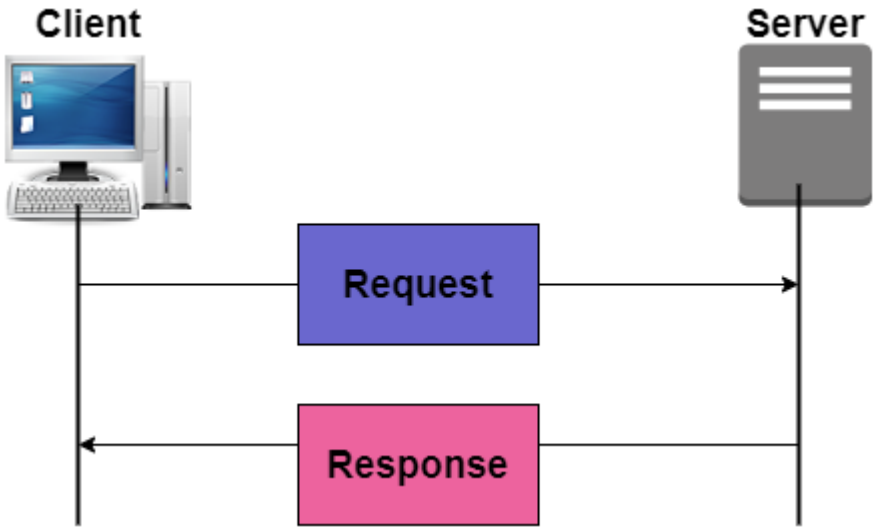
HTTP

HTTP stands for Hypertext Transfer Protocol and is mainly used to access the data on the world wide web i.e **(WWW)**. The **HTTP** mainly functions as the combination of **FTP**(File Transfer Protocol) and **SMTP**(Simple Mail Transfer Protocol).

- **HTTP** is one of the protocols used at the **Application Layer**.
- The **HTTP** is similar to **FTP** because **HTTP** is used to transfer the files and it mainly uses the services of **TCP.**
- Also, **HTTP** is much simpler than **FTP** because there is only **one TCP connection**.
- In **HTTP,** there is no separate control connection, as only data is transferred between the client and the server.
- The **HTTP** is like SMTP because the transfer of data between the client and server simply looks like **SMTP** messages. But there is a difference unlike **SMTP,** the **HTTP** messages are not destined to be read **by humans** as they are read and interpreted by **HTPP Client**(that is browser) and HTTP server.
- Also, **SMTP** messages are **stored and then forwarded** while the **HTTP** messages are **delivered immediately.**
- The **HTTP** mainly uses the services of the **TCP** on the well-known port that is **port 80.**
- **HTTP** is a **stateless protocol.**
- In **HTTP, the** client initializes the transaction by sending a request message, and the server replies by sending a response.
- This protocol is used to transfer the data in the form of plain text, hypertext, audio as well as video, and so on.

Working of HTTP

The HTTP makes use of Client-server architecture. As we have already told you that the browser acts as the HTTP client and this client mainly communicates with the webserver that is hosting the website.



The figure shows the HTTP transaction

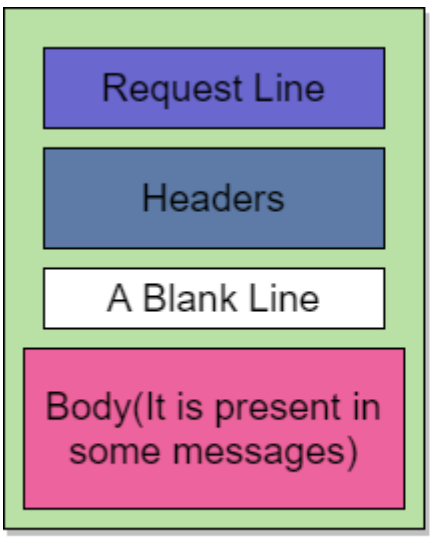
The format of the request and the response message is similar. The Request Message mainly consists of a request line, a header, and a body sometimes. A Response message consists of the status line, a header, and sometimes a body.

At the time when a client makes a request for some information (say client clicks on the hyperlink) to the webserver. The browser then sends a request message to the HTTP server for the requested objects.

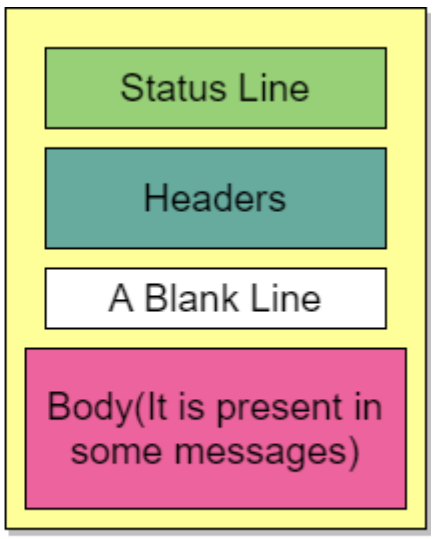
After that the following things happen:

- There is a connection that becomes open between the client and the webserver through the TCP.
- After that, the HTTP sends a request to the server that mainly collects the requested data.
- The response with the objects is sent back to the client by HTTP
- At last, HTTP closes the connection.

Let us take a look at the format of the request message and response message:



Request Message



Response Message

Request Line and Status line

The first line in the Request message is known as the request line, while the first line in the Response message is known as the Status line.

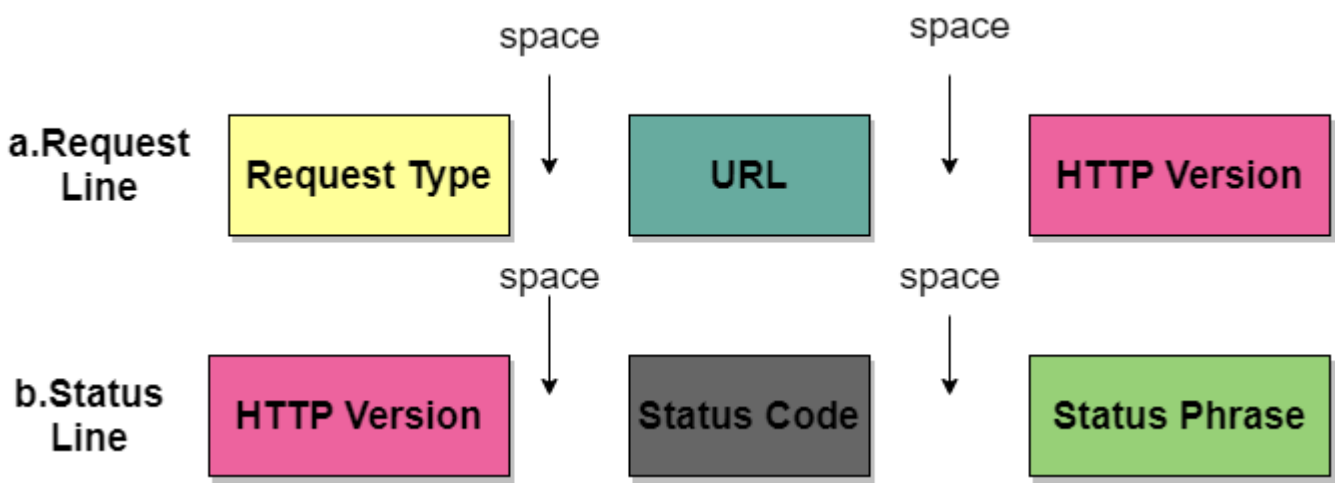


Figure: Request Line and Status Line

where,

Request Type

This field is used in the request line. The are several request types that are defined and these are mentioned in the table given below;

Name of Method	**Actions**
GET	This method is used to request a document from the server.
HEAD	This method mainly requests information about a document and not the document itself
POST	This method sends some information from the client to the server.

Name of Method	Actions
PUT	This method sends a document from the server to the client.
TRACE	This method echoes the incoming request.
CONNECT	This method means reserved
OPTION	In order to inquire about the available options.

URL

URL is a Uniform Resource locator and it is mainly a standard way of specifying any kind of information on the Internet.

HTTP Version

The current version of the HTTP is 1.1.

Status Code

The status code is the field of the response message.The status code consists of three digits.

Status Phrase

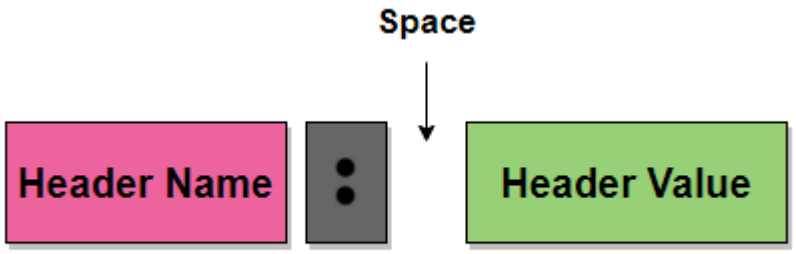
This field is also used in the response message and it is used to explain the status code in the form of text.

Header

The header is used to exchange the additional information between the client and the server. The header mainly consists of one or more header lines. Each header line has a header name, a colon, space, and a header value.

The header line is further categorized into four:

- **General Header** It provides general information about the message and it can be present in both request and response.
- **Request Header** It is only present in the request message and is used to specify the configuration of the client and the format of the document preferred by the client
- **Response Header** This header is only present in the response header and mainly specifies the configuration of the server and also the special information about the request.
- **Entity Header** It is used to provide information about the body of the document.



Body

It can be present in the request message or in the response message. The body part mainly contains the document to be sent or received.

Features of HTTP

The HTTP offers various features and these are as follows:

1. **HTTP is simple** The HTTP protocol is designed to be plain and human-readable.
2. **HTTP is stateless** Hypertext transfer protocol(HTTP) is a stateless protocol, which simply means that there is no connection among two requests that are being consecutively carried out on the same connection. Also, both the client and the server know each other only during the current requests and thus the core of the HTTP is itself a stateless one, On the other hand, the HTTP cookies provide in making use of stateful sessions.
3. **HTTP is extensible** The HTTP can be integrated easily with the new functionality by providing a simple agreement between the client and the server.
4. **HTTP is connectionless** As the HTTP request is initiated by the browser (HTTP client) and as per the request information by the user, after that the server processes the request of the client and then responds back to the client

Advantages of HTTP

Given below are the benefits of using HTTP:

1. There is no runtime support required to run properly.

2. As it is connectionless so there is no overhead in order to create and maintain the state and information of the session.
3. HTTP is usable over the firewalls and global application is possible.
4. HTTP is platform-independent.
5. HTTP reports the errors without closing the TCP connection.
6. Offers Reduced Network congestions.

Disadvanatges of HTTP

There are some drawbacks of using the HTTP protocol:

- HTTP is not optimized for mobile.
- HTTP is too verbose.
- It can be only used for point-to-point connections.
- This protocol does not have push capabilities.
- This protocol does not offer reliable exchange without the retry logic.

 The HTTP supports proxy servers. A proxy server is basically a computer that keeps the copies of the responses to recent requests. The proxy server mainly reduces the load on the originals server.In order to use the proxy server, the client must be configured in order to access the proxy instead of the target server.

HTTP Connections

HTTP connections can be further classified into two:

- Persistent Connection
- Nonpersistent Connection

Let us discuss them one by one:

1.Persistent Connection

In the persistent HTTP connection, all the requests and their corresponding responses are sent over the same TCP connections. The 1.1 version of the HTTP specifies a persistent connection by default.

In this type of connection, the server leaves the connection open for more requests after sending a response. Also, the server can close the connection at the request of the client or upon reaching the time-out.

In a Persistent connection, a single TCP connection is mainly used for sending multiple objects one after the other.

Usually, the length of the data is sent along with each response. There are some cases when the server does not know the length of the data this happens when the document is created dynamically and in such cases, the server informs the client that length is not known and closes the connection after sending the data so in order let the client Inform about the end of the data.

2.Nonpersistent Connection

In the Nonpersistent HTTP connection, one TCP connection is made for each request/response; it means there is a separate for each object.

Following are the steps used;

- The client opens a TCP connection and then sends a request.
- After that, the server sends the response and then closes the connection.
- Then the client reads the data and until it encounters an end-of-file marker then it closes the connection.

This connection imposes a high overhead on the server because N different buffers are required by the server, and the start procedure is slow each time when a connection is opened.

The nonpersistent connection is supported by the HTTP 1.0 version.

