Computer Networks:

Table of Contents

[What happens when either click an URL of type something in browser: 3](#_Toc165215074)

[TSP/IP Suite: 3](#_Toc165215075)

[OSI model (Open system interconnection) 4](#_Toc165215076)

[Physical Layer: 5](#_Toc165215077)

[Data Link Layer: 5](#_Toc165215078)

[Network: 5](#_Toc165215079)

[IPV4: 5](#_Toc165215080)

[IPV6: 8](#_Toc165215081)

[Difference between IPV4 and IPV6: 8](#_Toc165215082)

[Public Address and Private Address: 9](#_Toc165215083)

[NAT: 10](#_Toc165215084)

[ARP (Address Resolution Protocol-3389): 11](#_Toc165215085)

[ICMP: 11](#_Toc165215086)

[Debugging Tools: 11](#_Toc165215087)

[Ping: 11](#_Toc165215088)

[Traceroute: 12](#_Toc165215089)

[MTR (My TraceRoute): 12](#_Toc165215090)

[Nslookup: 12](#_Toc165215091)

[Unicasting: 13](#_Toc165215092)

[Multicasting: 14](#_Toc165215093)

[VPN: 14](#_Toc165215094)

[Transport Layer: 14](#_Toc165215095)

[Port 14](#_Toc165215096)

[Multiplexing and Demultiplexing: 14](#_Toc165215097)

[Flow Control: 14](#_Toc165215098)

[UDP: 14](#_Toc165215099)

[Error Control: 15](#_Toc165215100)

[Sliding Window: 15](#_Toc165215101)

[Congestion Control: 15](#_Toc165215102)

[UDP: User datagram protocol: 16](#_Toc165215103)

[Transmission Control Protocol (TCP): 16](#_Toc165215104)

[TCP 3-Way Handshake: 17](#_Toc165215105)

[Maximum segment size (MSS): 18](#_Toc165215106)

[TLS (Transport Layer Security): 18](#_Toc165215107)

[SSL (Secure Socket Layer): 18](#_Toc165215108)

[Application Layer: 19](#_Toc165215109)

[DHCP: (Dynamic Host Configuration Protocol- 67 and 68): 19](#_Toc165215110)

[DNS (Domain Name Service- 53): 19](#_Toc165215111)

[Telnet (23): 20](#_Toc165215112)

[SSH (Secure Shell-22): 20](#_Toc165215113)

[FTP (File transfer Protocol- 20 and 21): 20](#_Toc165215114)

[TFTP (Trivial file transfer Protocol- 69): 20](#_Toc165215115)

[URL: Uniform Resource locator: 20](#_Toc165215116)

[Common Gateway Interface: 20](#_Toc165215117)

[HTTP (Hypertext transfer protocol-80): 21](#_Toc165215118)

[Proxy Server: 21](#_Toc165215119)

[Reverse Proxy Server: 22](#_Toc165215120)

[HTTPS (443): 23](#_Toc165215121)

[SMTP (Simple mail Transfer Protocol-): 23](#_Toc165215122)

[DMARC: 24](#_Toc165215123)

[POP3 (Post Office Protocol-110): 24](#_Toc165215124)

[IMAP4 (Internet Mail Access Protocol): 24](#_Toc165215125)

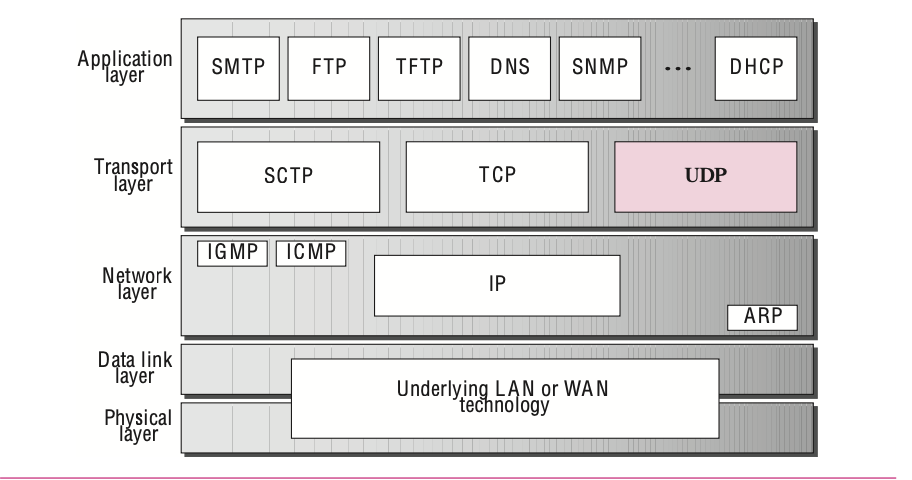
[Multipurpose Internet Mail Extensions (MIME): 24](#_Toc165215126)

# What happens when either click an URL of type something in browser:

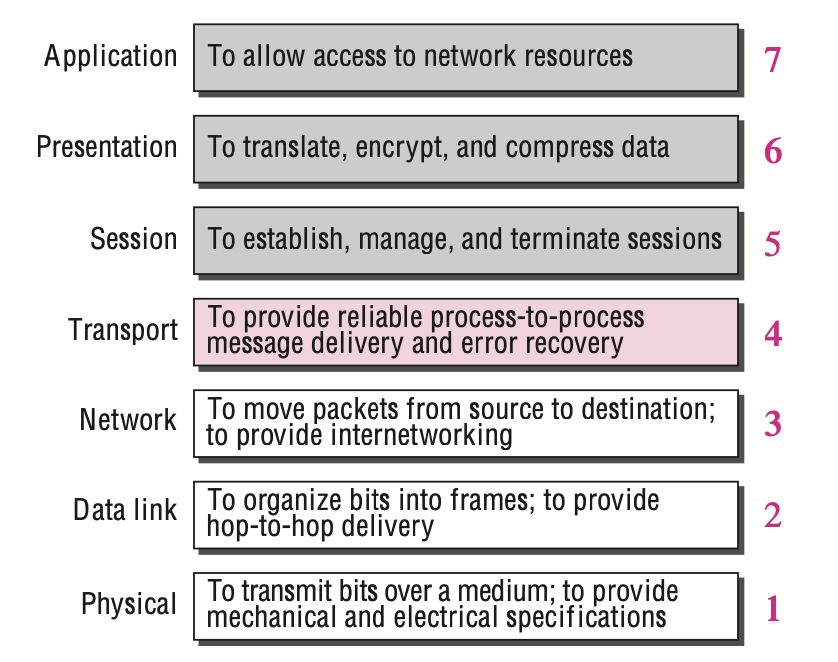
A screenshot of a computer screen

Description automatically generated

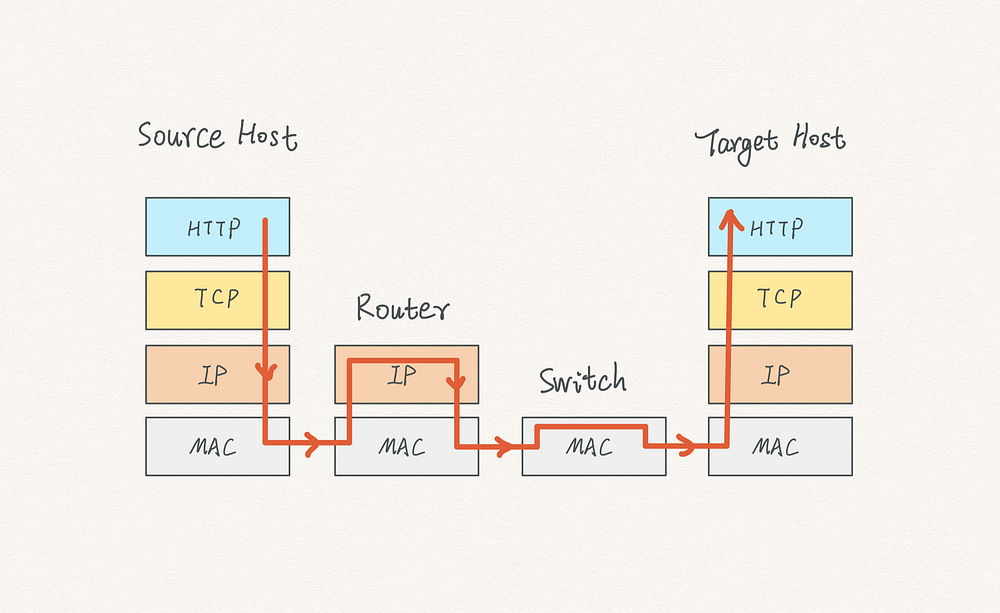
# TSP/IP Suite:



# OSI model (Open system interconnection)



TCP was designed based on the principles outlined in the OSI model, but it can operate independently of the OSI framework. However, the OSI model provides a conceptual foundation that aids in understanding how TCP and other networking protocols work together.



# Physical Layer:

* **The physical layer is responsible for moving individual bits from one (node) to the next.**
* The physical topology defines how devices are connected to make a network. Devices can be connected using a **mesh topology** (every device 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), or a **bus topology** (every device on a common link).
* The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full duplex. In the **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.
* The data link layer imposes a flow control mechanism to prevent overwhelming the receiver. The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames.
* 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:

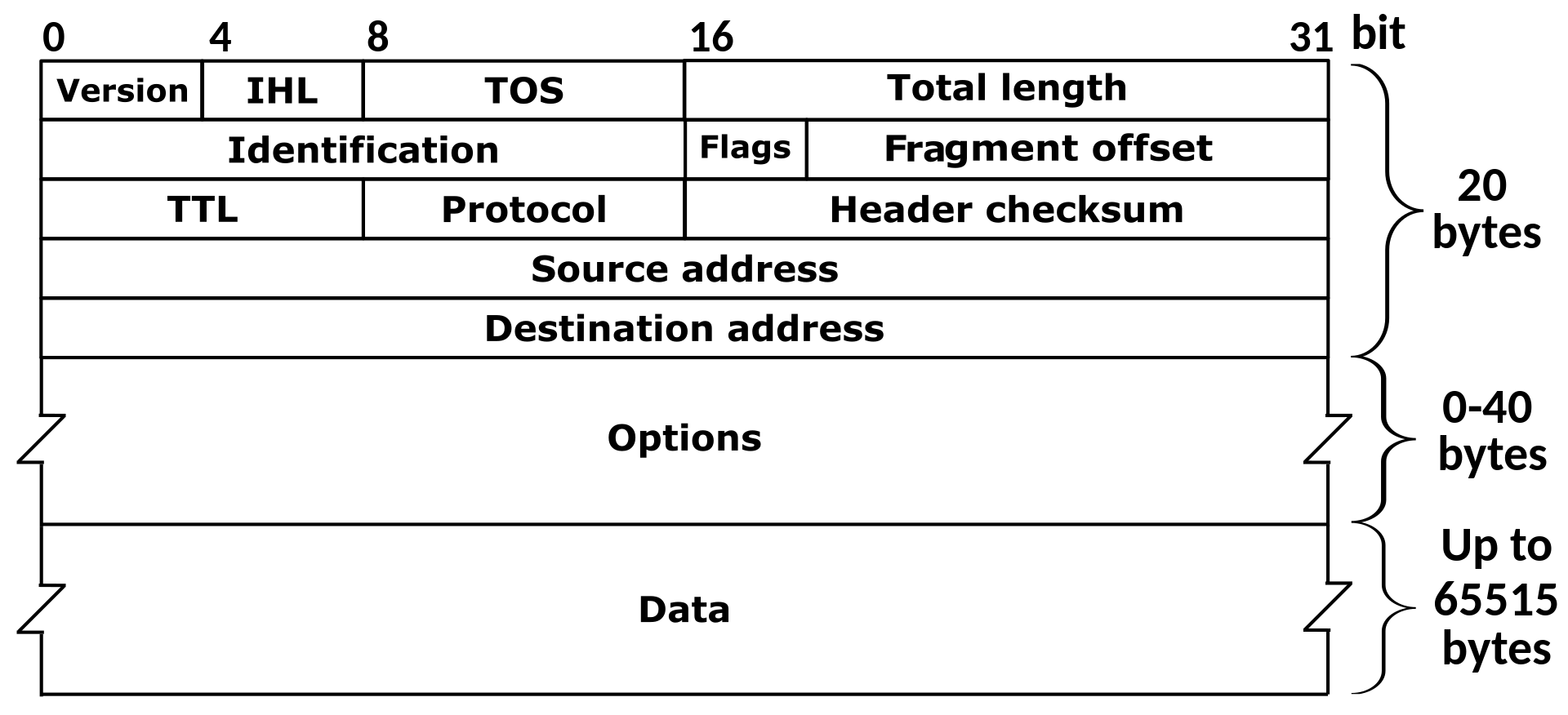
At this level, we are only concerned that a message from the application layer in one computer reaches the application layer in another computer. **In circuit switching, the whole message is sent from the source to the destination without being divided into packets.** The second solution to switching is called **packet switching**. The network layer in the Internet today is a packet-switched network. In this type of network, a message from the upper layer is divided into manageable packets and each packet is sent through the network. The source of the message sends the packets one by one; the destination of the message receives the packets one by one.

When the network layer provides a **connectionless service**, each packet traveling on the Internet is an independent entity; there is no relationship between packets belonging to the same message. The switches in this type of network are called *routers*. Each packet is routed based on the information contained in its header: source and destination address. The router in this case routes the packet based only on the destination address.

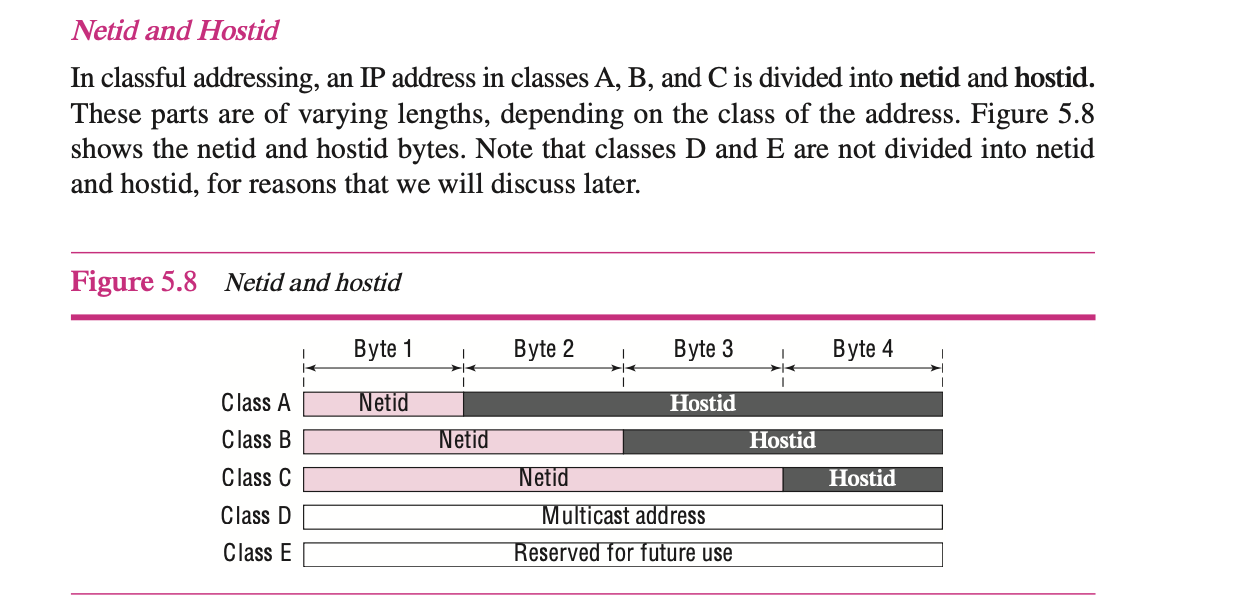
In a **connection-oriented service,** there is a relation between all packets belonging to a message. Each packet is forwarded based on the label in the packet. To create a connection-oriented service, a three-phase process is used: *setup*, *data transfer*, and *teardown.*

Since the network layer provides end-to-end communication, the two computers that need to communicate with each other each need a universal identification system, referred to as **network-layer address or logical address.**

## IPV4:



**An IPv4 address is 32 bits long. The IPv4 addresses are unique and universal. The address space of IPv4 is 232 or 4,294,967,296. The range of addresses allocated to an organization in classful addressing was a block of addresses in Class A, B, or C.**



**In classless addressing, the prefix defines the network, and the suffix defines the host. The prefix length in classless addressing can be 1 to 32. Total no of 1’s in the network mask is prefix.**

A graph with text and numbers

Description automatically generated with medium confidence

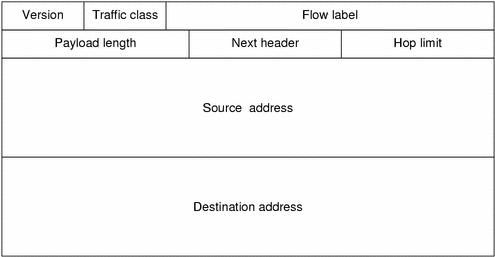
A close-up of a document

Description automatically generated

A screenshot of a computer

Description automatically generated

## IPV6:



## Difference between IPV4 and IPV6:

|  |  |
| --- | --- |
| **IPv4** | **IPv6** |
| IPv4 has a 32-bit address length | IPv6 has a 128-bit address length |
| It Supports Manual and DHCP address configuration | It supports Auto and renumbering address configuration |
| In IPv4 end to end, connection integrity is Unachievable | In IPv6 end-to-end, connection integrity is Achievable |
| It can generate 4.29×109 address space | The address space of IPv6 is quite large it can produce 3.4×1038address space |
| The Security feature is dependent on the application | IPSEC is an inbuilt security feature in the IPv6 protocol |
| Address representation of IPv4 is in decimal | Address Representation of IPv6 is in hexadecimal |
| Fragmentation performed by Sender and forwarding routers | In IPv6 fragmentation is performed only by the sender |
| In IPv4 Packet flow identification is not available | In IPv6 packet flow identification are Available and uses the flow label field in the header |
| In IPv4 checksum field is available | In IPv6 checksum field is not available |
| It has a broadcast Message Transmission Scheme | In IPv6 multicast and anycast message transmission scheme is available |
| In IPv4 Encryption and Authentication facility not provided | In IPv6 Encryption and Authentication are provided |
| IPv4 has a header of 20-60 bytes. | IPv6 has a header of 40 bytes fixed |
| IPv4 can be converted to IPv6 | Not all IPv6 can be converted to IPv4 |
| IPv4 consists of 4 fields which are separated by addresses dot (.) | IPv6 consists of 8 fields, which are separated by a colon (:) |
| IPv4’s  IP addresses are divided into five different classes. Class A , Class B, Class C, Class D , Class E. | IPv6 does not have any classes of the IP address. |
| IPv4 supports VLSM(Variable Length subnet mask). | IPv6 does not support VLSM. |
| Example of IPv4:  66.94.29.13 | Example of IPv6: 2001:0000:3238:DFE1:0063:0000:0000:FEFB |

## Public Address and Private Address:

| **Private IP Address** | **Public IP Address** |
| --- | --- |
| The scope of Private IP is local. | The scope of Public IP is global. |
| It is used to communicate within the network. | It is used to communicate outside the network. |
| Private IP addresses of the systems connected in a network differ in a uniform manner. | Public IP may differ in a uniform or non-uniform manner. |
| It works only on LAN. | It is used to get internet service. |
| It is used to load the network operating system. | It is controlled by ISP. |
| It is available free of cost. | It is not free of cost. |
| Private IP can be known by entering “ipconfig” on the command prompt. | Public IP can be known by searching “what is my ip” on Google. |
| **Range:**  ***10.0.0.0 – 10.255.255.255,***  ***172.16.0.0 – 172.31.255.255,***  ***192.168.0.0 – 192.168.255.255*** | **Range:** Besides private IP addresses, the rest are public. |
| Example: 192.168.1.10 | Example: 17.5.7.8 |
| Private IP uses numeric code that is not unique and can be used again | Public IP uses a numeric code that is unique and cannot be used by other |
| Private IP addresses are secure | The public IP address has no security and is  subjected to attack |
| Private IP addresses require NAT to communicate with devices | Public IP does not require a network translation |

NAT: The technology allows a site to use a set of private addresses for internal communication and a set of global Internet addresses (at least one) for communication with the rest of the world. All the outgoing packets go through the NAT router, which replaces the *source address* in the packet with the global NAT address. All incoming packets also pass through the NAT router, which replaces the *destination address* in the packet (the NAT router global address).

A diagram of a network

Description automatically generated

## ARP (Address Resolution Protocol-3389):

* The address resolution protocol (ARP) is a dynamic mapping method that finds a physical address given a logical address. An ARP request is broadcast to all devices on the network. An ARP reply is unicast to the host requesting the mapping. The ARP software package consists of five components: a cache table, queues, an output module, an input module, and a cache-control module.
* In proxy ARP, a router represents a set of hosts. When an ARP request seeks the physical address of any host in this set, the router sends its own physical address. This creates a subnetting effect. ATMARP is a protocol used on ATM networks that binds a physical address to an IP address.

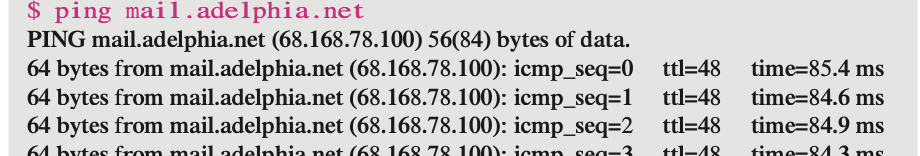
ICMP: The Internet Control Message Protocol (ICMP-) supports the unreliable and connectionless Internet Protocol (IP). ICMP messages are encapsulated in IP datagrams. There are two categories of ICMP messages: error-reporting and query messages. The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.

The code field for this type specifies the reason for discarding the datagram: **Destination-unreachable messages with codes 2 or 3 can be created only by the destination host. Other destination-unreachable messages can be created only by routers.**

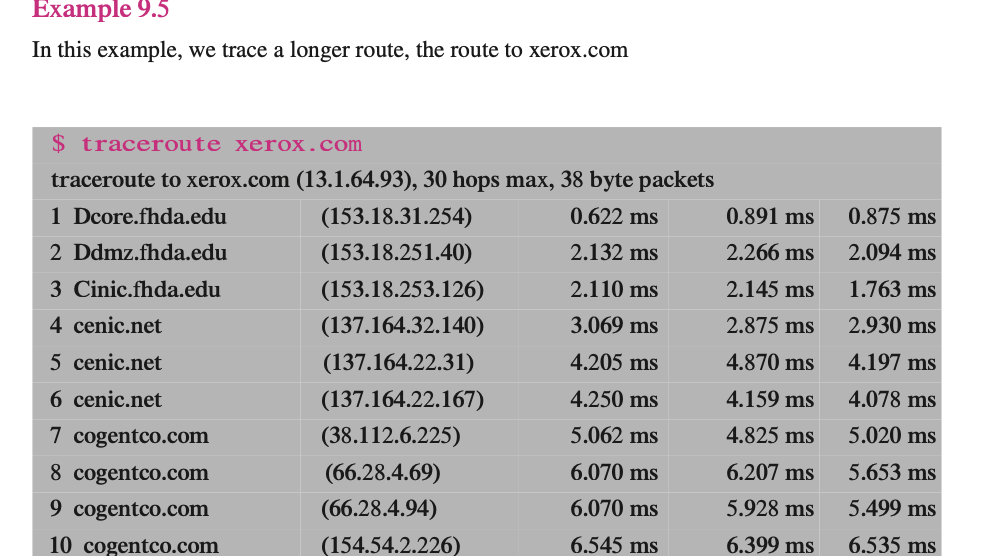
* **Code 0.** The network is unreachable, possibly due to hardware failure.
* **Code 1.** The host is unreachable. This can also be due to hardware failure.
* **Code 2.** The protocol is unreachable. An IP datagram can carry data belonging to higher-level protocols such as UDP, TCP, and OSPF.
* **Code 3.** The port is unreachable. The application program (process) that the data- gram is destined for is not running now.
* **Code 4.** Fragmentation is required, but the DF (do not fragment) field of the data- gram has been set. In other words, the sender of the datagram has specified that the datagram not be fragmented, but routing is impossible without fragmentation.
* **Code 5.** Source routing cannot be accomplished.
* **Code 6.** The destination network is unknown.
* **Code 7.** The destination host is unknown.
* **Code 8.** The source host is isolated.
* **Code 9.** Communication with the destination network is administratively prohibited.
* **Code 10.** Communication with the destination host is administratively prohibited.
* **Code 11.** The network is unreachable for the specified type of service.
* **Code 12.** The host is unreachable for the specified type of service. This is different from code 1. Here the router could route the datagram if the source had requested an available type of service.
* **Code 13.** The host is unreachable because the administrator has put a filter on it.
* **Code 14.** The host is unreachable because the host precedence is violated.
* **Code 15.** The host is unreachable because its precedence was cut off.

# Debugging Tools:

Ping: Used to find if the host is alive and responding. The source host sends ICMP echo request messages (type: 8, code: 0); the destination, if alive, responds with ICMP echo reply messages. the *ping* program sends messages with sequence numbers starting from 0. For each probe it gives us the RTT time. The TTL (time to live) field in the IP datagram that encapsulates an ICMP message has been set to 62, which means the packet cannot travel more than 62 hops.



Traceroute: The ***traceroute*** program in UNIX or ***tracert*** in Windows can be used to trace the route of a packet from the source to the destination



MTR (My TraceRoute): MTR is a combination of ping and traceroute that continuously monitors the network path between the local device and the destination host. It provides real-time statistics on latency and packet loss for each hop along the path, helping to identify network performance issues.

## Nslookup:

nslookup is a network utility program used to obtain information about Internet servers. As its name suggests, the utility finds name server information for domains by querying DNS. Nslookup is a command line driven utility supplied as part of most Windows operating systems that can reveal information related to domain names and the Internet Protocol (IP) addresses associated with them.

c:\nslookup (Press enter)

Default Server: sbs01.local.local (My default DNS Server)

Address: 192.168.16.2 (IP address of my default DNS Server)

> ? (find nslookup parameters instructions)

C:\>nslookup www.microsoft.com

Server: sbs01.local.local

Address: 192.168.16.2

Non-authoritative answer:

Name: lb1.www.ms.akadns.net

Addresses: 64.4.31.252

207.46.19.254

207.46.19.190

Aliases: www.microsoft.com

toggle.www.ms.akadns.net

g.www.ms.akadns.net

C:\>

The output above shows that nslookup has sent a request to my internal DNS server (sbs01.local.local located on 192.168.16.2) to action the request.

The answer that has been returned for my query was provided by the server called lb1.www.ms.akadns.net and it reports that three servers are known for www.microsoft.com and they are found at 64.4.31.252, 207.46.19.254 and 207.46.19.190.

Nslookup can also perform reverse lookups against DNS .ptr records. If we take one of the addresses returned in the previous example, you can see the true names of the servers used.

C:\>nslookup 207.46.19.190

Server: sbs01.local.local

Address: 192.168.16.2

Name: wwwbaytest1.microsoft.com

Address: 207.46.19.190

This reveals that server at 207.46.19.190 is called wwwbaytest1.microsoft.com is ONE of the servers that will respond to [www.microsoft.com](http://www.microsoft.com).

NC: Netcat

to check if port 80 is open on a server with IP address 192.168.1.100, you would use:

nc -zv 192.168.1.100 80

**netstat:** is a command-line utility in Linux and Unix-like operating systems used to display network-related information such as network connections, routing tables, interface statistics, masquerade connections, and multicast memberships. It provides detailed information about network connections and network interfaces on the system.

Some common options for the netstat command include:

* -a: Display all sockets (both listening and non-listening).
* -t: Display TCP connections.
* -u: Display UDP connections.
* -n: Show numerical addresses instead of resolving hostnames.
* -r: Display the routing table.
* -p: Show PID and program name for each connection.
* -l: Display only listening sockets.

**Nmap**: is a powerful open-source network scanning tool used to discover hosts and services on a computer network, thus creating a "map" of the network. It's commonly used for network inventory, security auditing, and vulnerability assessment.

Some common scan types for the nmap command include:

* -sS: TCP SYN scan (stealth scan).
* -sT: TCP connect scan.
* -sU: UDP scan.
* -sN, -sF, -sX: TCP Null, FIN, and Xmas scans (stealth scans).
* -sA: TCP ACK scan.
* -sP: Ping scan (detects live hosts without scanning ports).
* -sV: Version detection (detects services and their versions).
* -A: Aggressive scan (enables OS detection, version detection, script scanning, and traceroute).

Some common options for the nmap command include:

* -p port range: Specifies port(s) to scan.
* -O: Enable OS detection.
* -v: Increase verbosity (show more detailed output).
* -T timing: Set timing template (from 0 to 5 for timing options).
* -oN file: Save output in normal format to a file.
* -oX file: Save output in XML format to a file.

nmap is a versatile tool with numerous features and options, making it invaluable for network administrators, security professionals, and enthusiasts alike for network exploration and security auditing. However, it's important to use nmap responsibly and ethically, respecting the privacy and security of network hosts.

Unicasting: In *unicasting,* there is one source and one destination network. The relationship between the source and the destination network is one to one. Each router in the path of the datagram tries to forward the packet to one and only one of its interfaces.

Multicasting: In *multicasting,* there is one source and a group of destinations. The relationship is one to many. In this type of communication, the source address is a unicast address, but the destination address is a group address, a group of one or more destination networks in which there is at least one member of the group that is interested in receiving the multi- cast datagram.

# VPN:

One of the applications of Ipsec. It is a private but virtual network which ensure the privacy of an organisation. It is virtual because it doesn’t use real private WAN’s; the n/w is physically public but virtually private. VPN technology uses ESP (encapsulate secure payload; that provides source authentication, integrity, and confidentiality ) protocol of IPSec in the tunnel mode.

# Transport Layer:

Primary responsibility of this layer is to provide a process to process (An application layer entity or running program) communication. To send a message from one process to another, the transport layer protocol encapsulates and decapsulates messages. The packets at the transport layers on the Internet are called *user datagrams*, *segments*, or *packets*.

Port: Ports are necessary to make the multiple n/w request or to make multiple service available. Network connections are two ports: an open port listening on server and a randomly selected port on client’s computer. The local host and the remote host are defined using IP addresses. To define the pro- cesses, we need second identifiers called **port numbers.** In the TCP/IP protocol suite, the port numbers are integers between 0 and 65,535. If the request made from different client’s server distinguish the request based on client’s IP address. However, if the request is made from same client; server distinguish the requests based on port numbers. That’s whenever we open a new tab to make a request each tab gets assigned a new port number. In UNIX, the well-known ports are stored in a file called /etc/service.

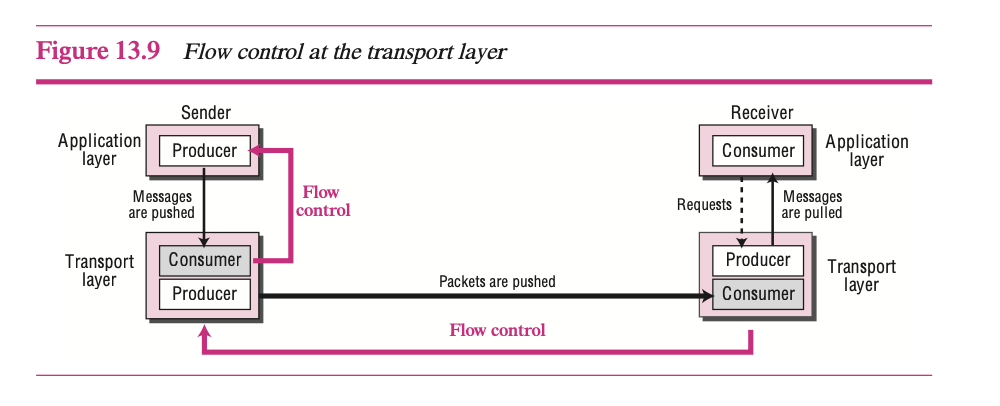
Difference b/w IP address and Port: The destination IP address defines the host among the different hosts in the world. After the host has been selected, the port number defines one of the processes on this host.

The combination of an IP address and a port number is called a **socket address.**

Multiplexing and Demultiplexing: Whenever an entity accepts items from more than one source, it is referred to as **multiplexing** (many to one); whenever an entity delivers items to more than one source, it is referred to as **demultiplexing** (one to many). The transport layer at the source performs multiplexing; the transport layer at the destination performs demultiplexing.

Flow Control: In communication at the transport layer, we are dealing with four entities: sender pro- cess, sender transport layer, receiver transport layer, and receiver process. The sending process at the application layer is only a producer. It produces message chunks and pushes them to the transport layer. The sending transport layer has a double role: it is both a consumer and the producer. It consumes the messages pushed by the producer. It encapsulates the messages in packets and pushes them to the receiving transport layer. The receiving transport layer has also a double role: it is the consumer for the packets received from the sender. It is also a producer; it needs to decapsulate the messages and deliver them to the application layer. Although flow control can be implemented in several ways, one of the solutions is normally to use two *buffers*. One at the sending transport layer and the other at the receiving transport layer. A buffer is a set of memory locations that can hold packets at the sender and receiver. The flow control communication can occur by sending signals from the consumer to producer.

## UDP:



## Error Control:

On the Internet, since the underlying network layer (IP), which is responsible to carry the packets from the sending transport layer to the receiving transport layer, is unreliable, we need to make the transport layer reliable if the application requires reliability. Reliability can be achieved to add error control service to the transport layer. Error control at the transport layer is responsible to

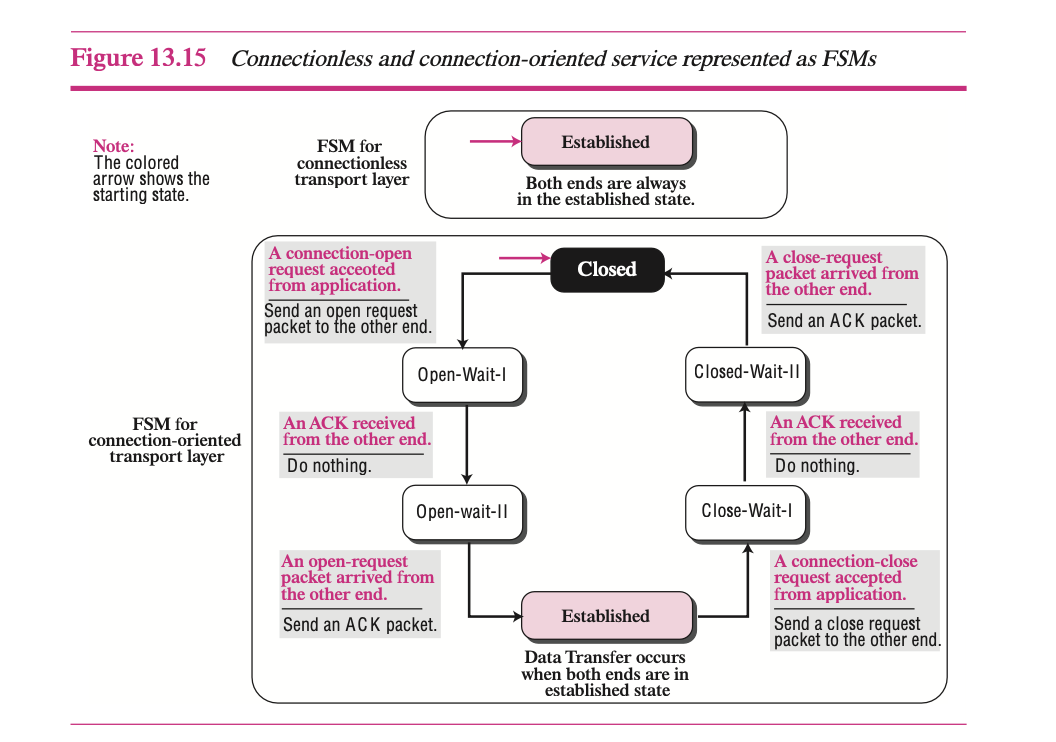
* Detect and discard corrupted packets.
* Keep track of lost and discarded packets and resend them.
* Recognize duplicate packets and discard them.
* Buffer out-of-order packets until the missing packets arrive.

Error control, unlike the flow control, involves only the sending and receiving transport layers. So, we add a sequence number field in a header to manage the error control on layer.

Sliding Window: The buffer is represented as a set of slices, called the **sliding window,** that occupy part of the circle at any time. At the sender site, when a packet is sent, the corresponding slice is marked. When all the slices are marked, it means that the buffer is full, and no further messages can be accepted from the application layer.

Congestion Control: Congestion in a network may occur if the **load** on the network—the number of packets sent to the network—is greater than the *capacity* of the network—the number of packets a network can handle. **Congestion control** refers to the mechanisms and techniques to control the congestion and keep the load below the capacity. Congestion happens in any system that involves waiting.

FSM: Finite State Machine for connectionless and connection-oriented network:



## UDP: User datagram protocol:

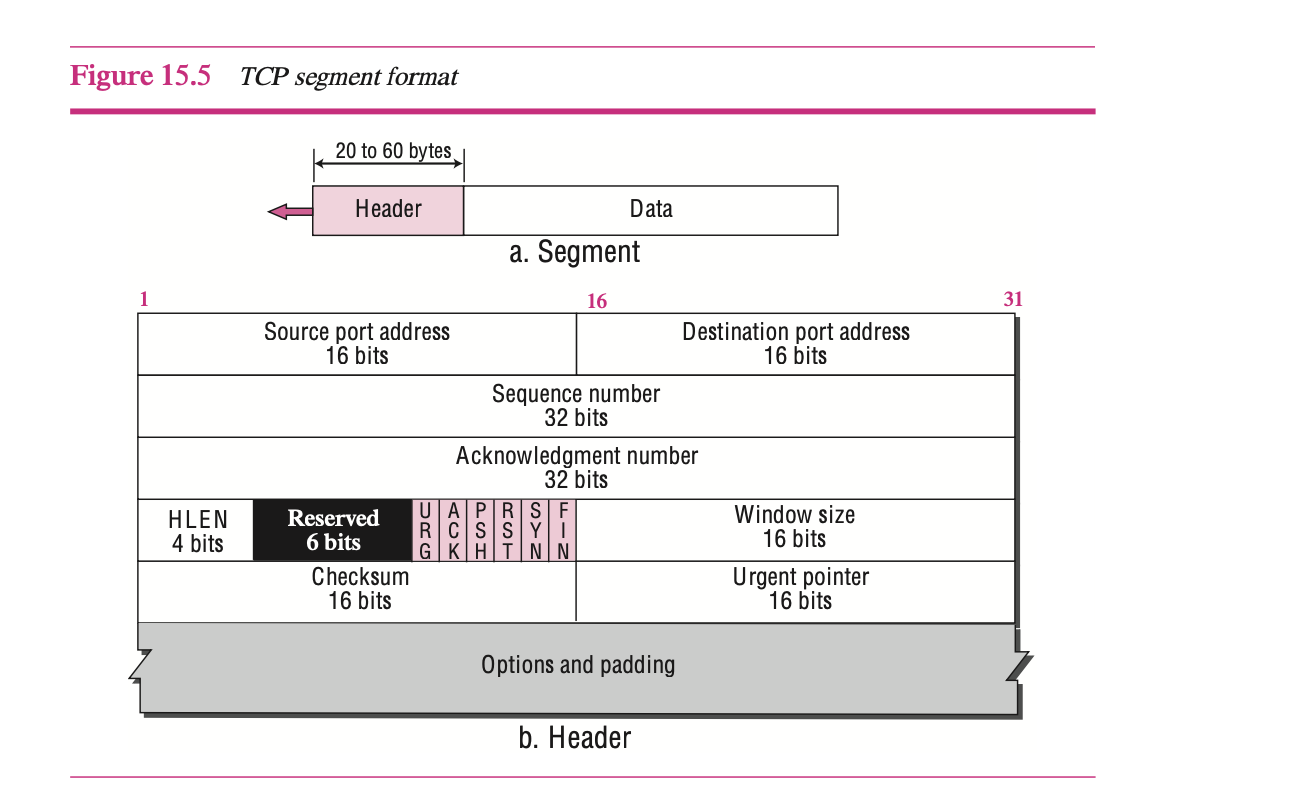
* UDP is a (mostly) unreliable and connectionless protocol that requires little overhead and offers fast delivery.
* UDP packets, called **user datagrams,** have a fixed-size header of 8 bytes.
* UDP’s only attempt at error control is the checksum. Inclusion of a pseudo header in the checksum calculation allows source and destination IP address errors to be detected. UDP has no flow-control and congestion control mechanism.

A diagram of a computer program

Description automatically generated with medium confidence

## Transmission Control Protocol (TCP):

* TCP provides process-to-process, full-duplex, and connection-oriented service.
* The unit of data transfer between two devices using TCP software is called a segment; it has 20 to 60 bytes of header, followed by data from the application program.
* A TCP connection consists of three phases: connection establishment, data trans- fer, and connection termination.
* Connection establishment requires three-way handshaking; connection termination requires three- or four-way handshaking.
* TCP uses flow control, implemented as a sliding window mechanism. CPuseserrorcontroltoprovideareliableservice.Errorcontrolishandledbycheck- sums, acknowledgment, and time-outs.
* TCP uses four timers (retransmission, persistence, keepalive, and time-wait) in its operation. In TCP, there can be only be one RTT (Round trip time) measurement in progress at any time. TCP does not consider the RTT of a retransmitted segment in its calculation of an RTT.
* MSS is determined during connection establishment. Each party defines the MSS for the segments it will receive during the connection. If a party does not define this, the default values is 536 bytes.



TCP 3-Way Handshake: It is a crucial process for establishing a connection between a client and a server. It involves the exchange of information, including initial sequence numbers (ISN) and parameters like window scale factor and Maximum Segment Size.

The handshake consists of three steps:

* The client sends an SYN message, indicating the initiation of the connection and providing its initial sequence number and parameters.
* The server responds with an SYN/ACK message, acknowledging the client's request, providing its own initial sequence number, and confirming the receipt of the client's message.
* The client sends an ACK message, acknowledging the server's response and completing the handshake.

The initial sequence number is a random one, and it cannot be zero. the server’s ISN is a random number and

different from the client’s initial one. server's ACK sequence number = 1 + client's initial sequence number. the client's ACK sequence number = 1 + the server's initial sequence number.

During the handshake, TCP transitions through different states:

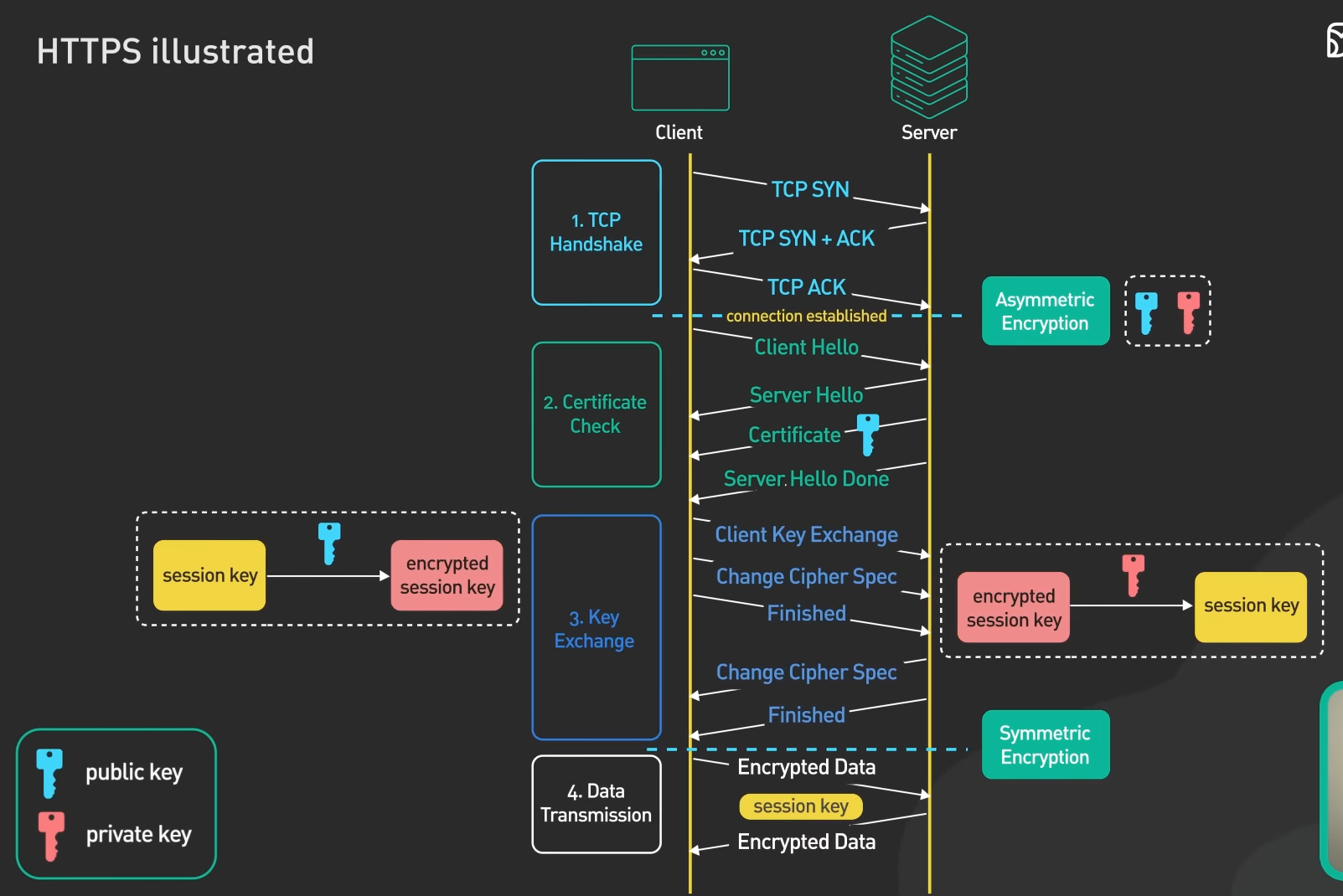
* Initially, both ends are in the Closed state.
* The server transitions to the Listen state when configured to listen on specific ports.
* When the client sends the SYN message, its TCP transitions to the SYN-Sent state, while the server's TCP enters the SYN-Received state upon receiving the SYN message.
* After the server sends the SYN/ACK message, the client's TCP transitions to the Established state and sends the final ACK message.
* Upon receiving the ACK message, the server's TCP also enters the Established state.

However, attackers may exploit the handshake process by initiating SYN messages without sending the final ACK, causing the server's TCP connections to remain stuck in the SYN-Received state and potentially exhausting server resources, hindering legitimate connections.

## Maximum segment size (MSS):

The **maximum-segment-size option** defines the size of the biggest unit of data that can be received by the destination of the TCP segment. Despite its name, it defines the maximum size of the data, not the maximum size of the segment. Since the field is 16 bits long, the value can be 0 to 65,535 bytes. MSS is determined during connection establishment. Each party defines the MSS for the segments it will receive during the connection. If a party does not define this, the default values is 536 bytes.

## TLS (Transport Layer Security):

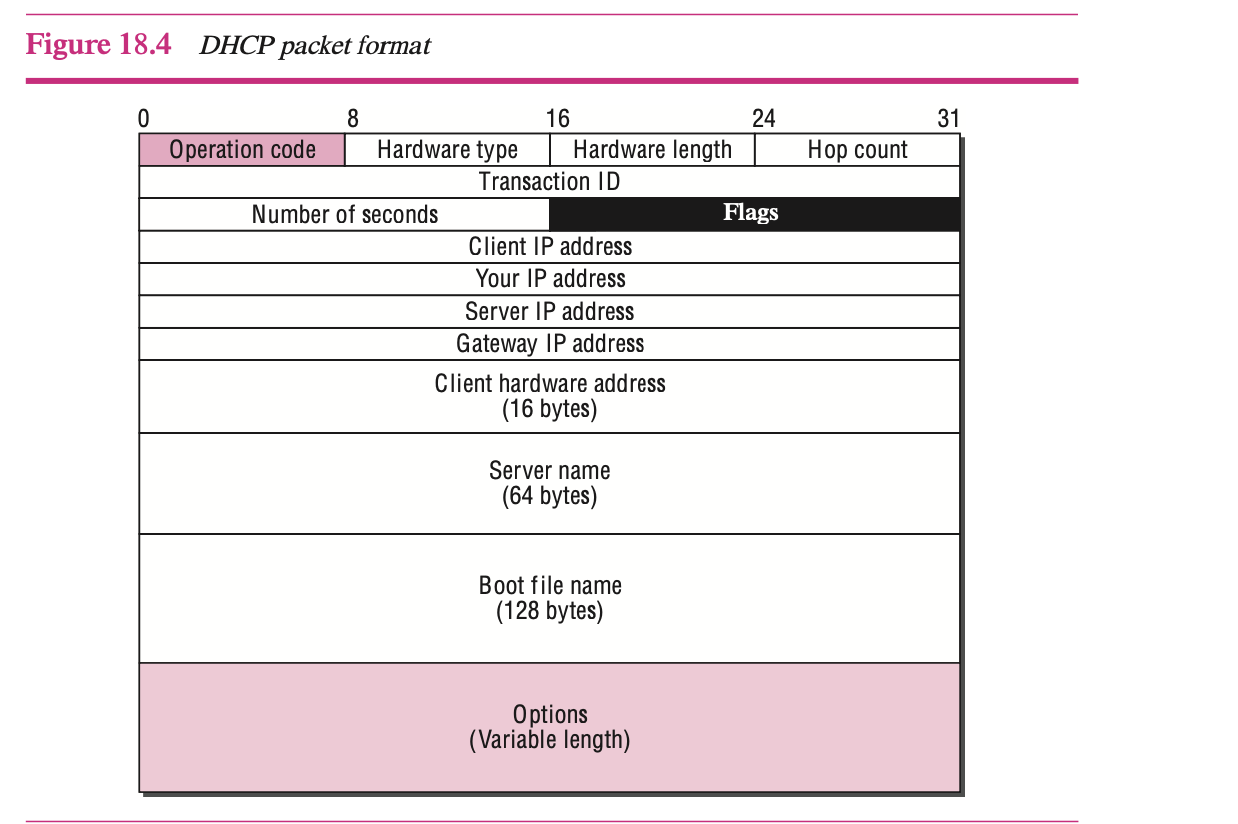


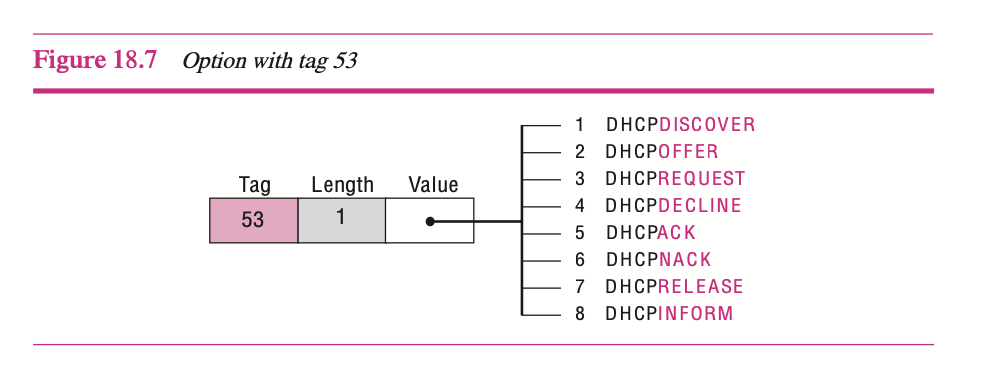
## SSL (Secure Socket Layer):

# Application Layer:

## DHCP: (Dynamic Host Configuration Protocol- 67 and 68):

The **Dynamic Host Configuration Protocol (DHCP)** is a client/server protocol designed to provide the four pieces of information for a diskless computer or a computer that is booted for the first time. It is used for dynamic IP addressing. It provides the IP address to a device on lease bases to make sure that, server doesn’t get out of IP. It can also provide the parament IP to some specific devices like printer, router etc based upon their MAC address.





## 

## DNS (Domain Name Service- 53):

The Domain Name System (DNS) is a client-server application that identifies each host on the Internet with a unique user-friendly name. DNS organizes the name space in a hierarchical structure to decentralize the responsibilities involved in naming. DNS can be pictured as an inverted hierarchical tree structure with one root node at the top and a maximum of 128 levels. Each node in the tree has a domain name. A domain is defined as any subtree of the domain name space.

The domain name space is divided into three sections: generic domains, country domains, and inverse domain. There are two types of DNS messages: queries and responses. There are two types of DNS records: question records and resource records. DNS uses an offset pointer for duplicated domain name information in its messages. Dynamic DNS (DDNS) automatically updates the DNS master file. DNS uses the services of UDP for messages of less than 512 bytes; otherwise, TCP is used.

To protect DNS, IETF has devised a technology named DNS Security (DNSSEC) that provides the message origin authentication and message integrity using a security service called digital signature.

## Telnet (23):

TELNET is a general-purpose client-server application program. TELNET is a client-server application that allows a user to log on to a remote machine, giving the user access to the remote system. TELNET uses the Network Virtual Terminal (NVT) system to encode characters on the local system.

## SSH (Secure Shell-22):

SSH is more secure and provide more services than telnet.

## FTP (File transfer Protocol- 20 and 21):

It is std mechanism provided by TCP/IP to transfer the file from one host to another and uses the services of TCP. It needs two TCP connections. The well-known port 21 is used for the control connection and the well-known port 20 for the data connection.

There are six classes of commands sent by the client to establish communication with the server: access commands, file management commands, data formatting commands, port defining commands, file transferring commands, and miscellaneous commands. There are three types of file transfer: server-to-client file transfer, client-to-server file transfer, transfer of list of directories.

Transferring files with FTP is not secure. One solution to provide security is to add a Secure Socket Layer (SSL) between the FTP application layer and the TCP layer.

## TFTP (Trivial file transfer Protocol- 69):

TFTP is very useful for basic file transfer where security is not a big issue. It can be used to initialize devices such as bridges or routers. Its main application is in conjunction with the DHCP. TFTP requires only a small amount of memory and uses only the services of UDP(Port 69) and IP. It can easily be configured in ROM (or PROM). A client uses the services of TFTP to retrieve a copy of a file or send a copy of a file to a server. (At the time of re-booting the system).

## URL: Uniform Resource locator:

A client that wants to access a Web page needs the file name and the address. To facilitate the access of documents distributed throughout the world, HTTP uses locators. The **uniform resource locator (URL)** is a standard locator for specifying any kind of information on the Internet.

URL= Protocol:// Host: Port/path

## Common Gateway Interface:

The **Common Gateway Interface (CGI)** is a technology that creates and handles dynamic documents. CGI is a set of standards that defines how a dynamic document is written, how data are input to the program, and how the output result is used.

CGI is not a new language; instead, it allows programmers to use any of several languages such as C, C++, Bourne Shell, Korn Shell, C Shell, Tcl, or Perl. The only thing that CGI defines is a set of rules and terms that the programmer must follow.

The term *common* in CGI indicates that the standard defines a set of rules that is common to any language or platform. The term *gateway* here means that a CGI pro- gram can be used to access other resources such as databases, graphic packages, and so on.

## HTTP (Hypertext transfer protocol-80):

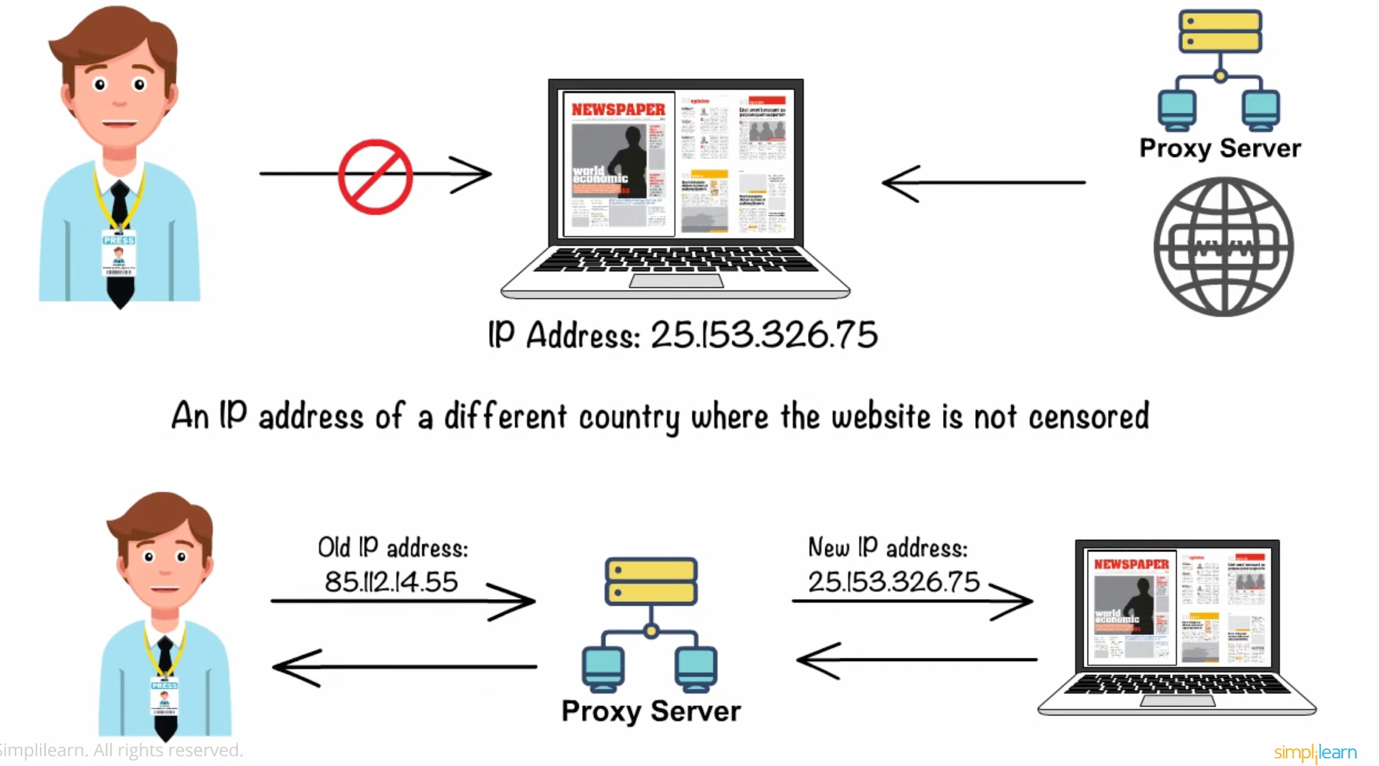
* The Hypertext Transfer Protocol (HTTP) is the main protocol used to access data on the World Wide Web (WWW). HTTP uses a TCP connection to transfer files. HTTP transactions are made of request and response messages.
* HTTP can be used in two modes: nonpersistent and persistent. The nonpersistent mode uses a new TCP connection for each transaction; the persistent mode uses only one connection. The default in the new version of HTTP is the persistent mode.
* HTTP can use cookies to keep the state of the transactions. The server sends a cookie that can be stored in the client and be retrieved later by the server.

## Proxy Server:

HTTP supports **proxy servers.** A proxy server is a computer that keeps copies of responses to recent requests. The HTTP client sends a request to the proxy server. The proxy server checks its cache. If the response is not stored in the cache, the proxy server sends the request to the corresponding server. Incoming responses are sent to the proxy server and stored for future requests from other clients.

The proxy server reduces the load on the original server, decreases traffic, and improves latency. However, to use the proxy server, the client must be configured to access the proxy instead of the target server.

Note that the proxy server acts both as a server and client. The proxy servers are normally located at the client site.



## Reverse Proxy Server:

A reverse proxy server is a type of proxy server that typically sits behind the firewall in a private network and directs client requests to the appropriate back-end server. A reverse proxy provides an additional level of abstraction and control to ensure the smooth flow of network traffic between clients and servers.

Common uses for a reverse proxy server include:

Load balancing – A reverse proxy server can act as a “traffic cop,” sitting in front of your backend servers and distributing client requests across a group of servers in a manner that maximizes speed and capacity utilization while ensuring no one server is overloaded, which can degrade performance. If a server goes down, the load balancer redirects traffic to the remaining online servers.

Web acceleration – Reverse proxies can compress inbound and outbound data, as well as cache commonly requested content, both of which speed up the flow of traffic between clients and servers. They can also perform additional tasks such as SSL encryption to take load off of your web servers, thereby boosting their performance.

Security and anonymity – By intercepting requests headed for your backend servers, a reverse proxy server protects their identities and acts as an additional defense against security attacks. It also ensures that multiple servers can be accessed from a single record locator or URL regardless of the structure of your local area network.

Reverse proxies can hide the existence and characteristics of an origin server or servers.

Application firewall features can protect against common web-based attacks, such as DoS or DDoS. Without a reverse proxy, removing malware or initiating takedowns, for example, can become difficult.

In the case of secure websites, a web server may not perform SSL encryption itself, but instead offloads the task to a reverse proxy that may be equipped with SSL acceleration hardware. (See SSL termination proxy.)

A reverse proxy can distribute the load from incoming requests to several servers, with each server serving its own application area. In the case of reverse proxying in the neighbourhood of web servers, the reverse proxy may have to rewrite the URL in each incoming request in order to match the relevant internal location of the requested resource.

A reverse proxy can reduce load on its origin servers by caching static content, as well as dynamic content - also known as web acceleration. Proxy caches of this sort can often satisfy a considerable number of website requests, greatly reducing the load on the origin server(s). A reverse proxy can optimize content by compressing it to speed up loading times.

In a technique known as "spoon-feed"[2] a dynamically generated page can be produced all at once and served to the reverse-proxy, which can then return it to the client a little bit at a time. The program that generates the page need not remain open, thus releasing server resources during the possibly extended time the client requires to complete the transfer.

Reverse proxies can operate wherever multiple webservers must be accessible via a single public IP address. The web servers listen on different ports in the same machine, with the same local IP address or, possibly, on different machines and different local IP addresses altogether. The reverse proxy analyses each incoming request and delivers it to the right server within the local area network.

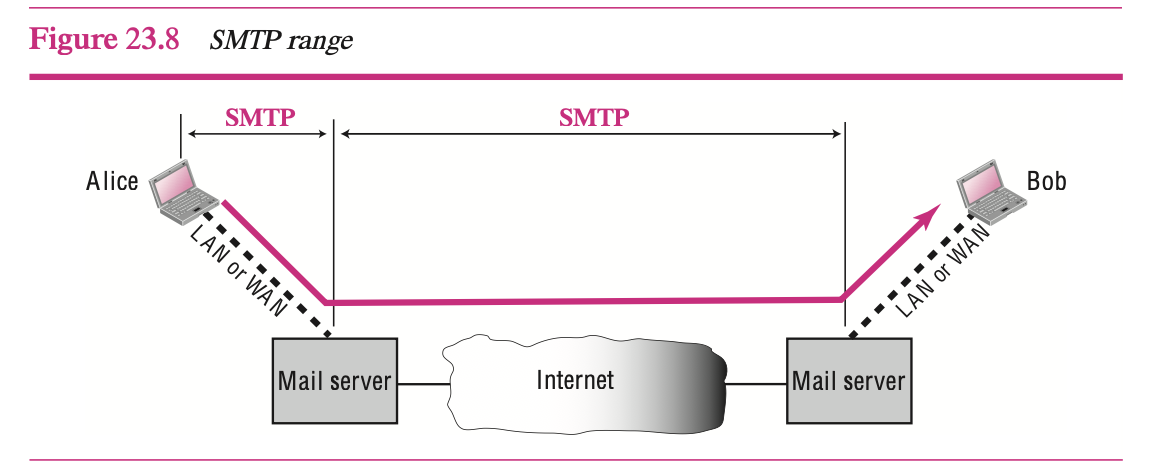
Reverse proxies can perform A/B testing and multivariate testing without placing JavaScript tags or code into pages.

Commercial or enterprise level out-of-box solutions exist and can have an agent installed on user systems to ensure a constant connection to a cloud proxy / reverse proxy server also a SaaS solution. An example of this is McAfee Web Protection Gateway - SaaS solution.

A reverse proxy can add basic HTTP access authentication to a web server that does not have any authentication.

## HTTPS (443):

SMTP (Simple mail Transfer Protocol-): The actual mail transfer is done through message transfer agents (MTAs). 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 **Simple Mail Transfer Protocol (SMTP).** SMTP is used two times, between the sender and the sender’s mail server and between the two mail servers. The process of transferring a mail message occurs in three phases: **connection establishment,** mail transfer, and **connection termination.**



## DMARC:

## POP3 (Post Office Protocol-110):

**Post Office Protocol, version 3 (POP3)** is simple and limited in functionality. Mail access starts with the client when the user needs to download its 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 username and password to access the mailbox. 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.

## IMAP4 (Internet Mail Access Protocol):

IMAP4 provides the following extra functions:

* A user can check the e-mail header prior to downloading.
* A user can search the contents of the e-mail for a specific string of characters prior to downloading.
* A user can partially download e-mail. This is especially useful if bandwidth is limited, and the e-mail contains multimedia with high bandwidth requirements.
* A user can create, delete, or rename mailboxes on the mail server.
* A user can create a hierarchy of mailboxes in a folder for e-mail storage.

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 it to the client MTA to be sent through the Internet. The message at the receiving site is transformed back to the original data. Secure e-mail is possible through two technologies: Good Privacy (PGP) and SMIME (Secure MIME)