ITPC405 COMPUTER NETWORK

UNIT IV

Topics

Transport Layer

- **Functions of Transport Layer**

- **User Datagram Protocol (UDP)**
  - **Applications using UDP**

- **Transmission Control Protocol (TCP)**
  - **Services provided by the TCP**
  - **Connection Establishment in TCP**

- **UDP vs TCP**

- **Congestion Control**
  - **Leaky Bucket algorithm**
  - **Token Bucket algorithm**

- **Quality of Service**

Transport Layer

The basic function of the Transport layer is to accept data from the layer above, split it up into smaller units, pass these data units to the Network layer, and ensure that all the pieces arrive correctly at the other end. The Transport layer also determines what type of service to provide to the Session layer, and, ultimately, to the users of the network. The Transport layer is a true end-to-end layer from the source to the destination. In other words, a program on the source machine carries on a conversation with a similar program on the destination machine, using the message headers and control messages.

Functions of Transport Layer

1. **Service Point Addressing**: Transport Layer header includes service point address which is port address. This layer gets the message to the correct process on the computer unlike Network Layer, which gets each packet to the correct computer.
2. **Segmentation and Reassembling**: A message is divided into segments; each segment contains sequence number, which enables this layer in reassembling the message. Message is reassembled correctly upon arrival at the destination and replaces packets which were lost in transmission.
3. **Connection Control**: It includes 2 types:
    - Connectionless Transport Layer: Each segment is considered as an independent packet and delivered to the transport layer at the destination machine.
    - Connection Oriented Transport Layer: Before delivering packets, connection is made with transport layer at the destination machine.
4. **Flow Control**: In this layer, flow control is performed end to end.
5. **Error Control**: Error Control is performed end to end in this layer to ensure that the complete message arrives at the receiving transport layer without any error. Error Correction is done through retransmission.

Most common Transport Protocols that are used on the Internet are

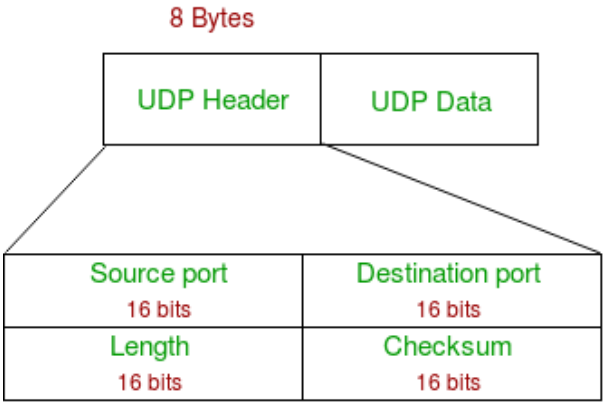
1. User Datagram Protocol (UDP)
2. Transmission Control Protocol (TCP)

User Datagram Protocol (UDP)

User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of Internet Protocol suite, referred as UDP/IP suite. It is unreliable and connectionless protocol**.** It is an example of a connectionless protocol. UDP provides no guarantee mechanism. Because of this, UDP is ideal for real-time, streaming data transmissions, like voice and video-conferencing.

UDP header is **8-bytes** fixed and simple header. First 8 Bytes contains all necessary header information and remaining part consists of data. UDP port number fields are each 16 bits long, therefore range for port numbers defined from 0 to 65535; port number 0 is reserved. Port numbers help to distinguish different user requests or process.

UDP Header



UDP Header Format

- **Source Port :** Source Port is 2 Byte long field used to identify port number of source.
- **Destination Port :** It is 2 Byte long field, used to identify the port of destined packet.
- **Length :** Length is the length of UDP including header and the data. It is 16-bits field.
- **Checksum :** Checksum is 2 Bytes long field. It is the 16-bit one's complement of the one's complement sum of the UDP header, pseudo header of information from the IP header and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.

Applications using UDP

- Applications which require one response for one request use UDP. Example- DNS.
- Routing Protocols like RIP and OSPF use UDP because they have very small amount of data to be transmitted.
- Trivial File Transfer Protocol (TFTP) uses UDP to send very small sized files.
- Broadcasting and multicasting applications use UDP.
- Streaming applications like multimedia, video conferencing etc use UDP since they require speed over reliability.
- Real time applications like chatting and online games use UDP.
- Management protocols like SNMP (Simple Network Management Protocol) use UDP.
- Bootp / DHCP uses UDP.
- Other protocols that use UDP are- Kerberos, Network Time Protocol (NTP), Network News Protocol (NNP), Quote of the day protocol etc.

Transmission Control Protocol (TCP)

The Transmission Control Protocol is the most common transport layer protocol. It works together with IP and provides a reliable transport service between processes using the network layer service provided by the IP protocol.

The various **services** provided by the TCP to the application layer are:

Process-to-Process Communication

TCP provides process to process communication, i.e, the transfer of data takes place between individual processes executing on end systems. This is done using port numbers or port addresses. Port numbers are 16 bit long that help identify which process is sending or receiving data on a host.

Stream oriented

This means that the data is sent and received as a stream of bytes(unlike UDP or IP that divides the bits into datagrams or packets). However, the network layer, that provides service for the TCP, sends packets of information not streams of bytes. Hence, TCP groups a number of bytes together into a *segment* and adds a header to each of these segments and then delivers these segments to the network layer. At the network layer, each of these segments are encapsulated in an IP packet for transmission. The TCP header has information that is required for control purpose which will be discussed along with the segment structure.

Full duplex service

This means that the communication can take place in both directions at the same time.

Connection oriented service

Unlike UDP, TCP provides connection oriented service. It defines 3 different phases:

- Connection establishment
- Data transfer
- Connection termination

(This is a virtual connection, not a physical connection, means during the transmission the resources will not be reserved and the segments will not follow the same path to reach the destination but it is a connection orientation in the sense that segments will arrive in order by the help of sequence number.)

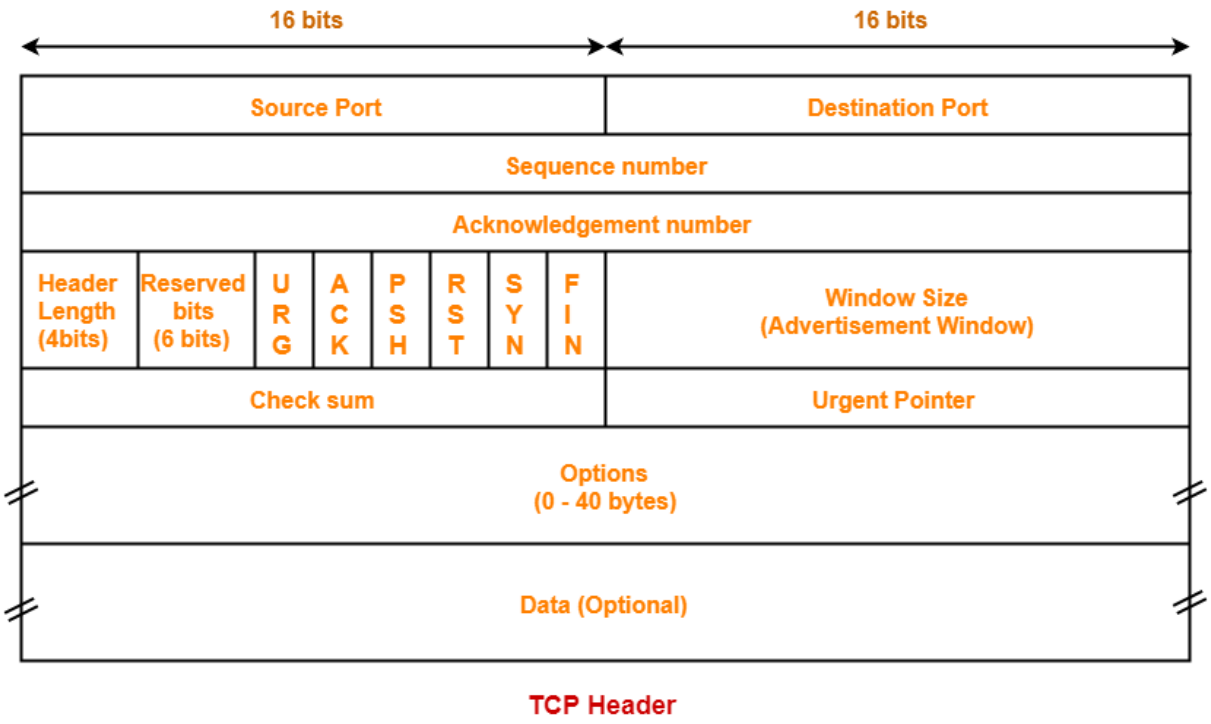
Reliability

TCP is reliable as it uses checksum for error detection, attempts to recover lost or corrupted packets by re-transmission, acknowledgement policy and timers. It uses features like byte number and sequence number and acknowledgement number so as to ensure reliability. Also, it uses congestion control mechanisms.

Multiplexing

TCP does multiplexing and de-multiplexing at the sender and receiver ends respectively as a number of logical connections can be established between port numbers over a physical connection.

TCP Header



TCP Header

Source Port

- Source Port is a 16 bit field.
- It identifies the port of the sending application.

Destination Port

- Destination Port is a 16 bit field.
- It identifies the port of the receiving application.

Sequence Number

- Sequence number is a 32 bit field.
- TCP assigns a unique sequence number to each byte of data contained in the TCP segment.
- This field contains the sequence number of the first data byte.

Acknowledgement Number

- Acknowledgment number is a 32 bit field.
- It contains sequence number of the data byte that receiver expects to receive next from the sender.
- It is always sequence number of the last received data byte incremented by 1.

Header Length

- Header length is a 4 bit field.
- It contains the length of TCP header.
- It helps in knowing from where the actual data begins.

Reserved Bits

- The 6 bits are reserved.
- These bits are not used.

URG Bit

When URG bit is set to 1,

- It indicates the receiver that certain amount of data within the current segment is urgent.
- Urgent data is pointed out by evaluating the urgent pointer field.
- The urgent data has be prioritized.
- Receiver forwards urgent data to the receiving application on a separate channel.

ACK Bit

- When ACK bit is set to 1, it indicates that acknowledgement number contained in the TCP header is valid.
- For all TCP segments except request segment, ACK bit is set to 1.
- Request segment is sent for connection establishment during **Three Way Handshake**.

PSH Bit

When PSH bit is set to 1,

- All the segments in the buffer are immediately pushed to the receiving application.
- No wait is done for filling the entire buffer.
- This makes the entire buffer to free up immediately.

RST Bit

When RST bit is set to 1,

- It indicates the receiver to terminate the connection immediately.
- It causes both the sides to release the connection and all its resources abnormally.
- The transfer of data ceases in both the directions.
- It may result in the loss of data that is in transit.

This is used only when-

- There are unrecoverable errors.
- There is no chance of terminating the TCP connection normally.

SYN Bit

When SYN bit is set to 1,

- It indicates the receiver that the sequence number contained in the TCP header is the initial sequence number.
- Request segment sent for connection establishment during Three way handshake contains SYN bit set to 1.

FIN Bit

When FIN bit is set to 1,

- It indicates the receiver that the sender wants to terminate the connection.
- FIN segment sent for TCP connection termination contains FIN bit set to 1.

Window Size

- Window size is a 16 bit field.
- It contains the size of the receiving window of the sender.
- It advertises how much data (in bytes) the sender can receive without acknowledgement.
- Thus, window size is used for Flow control.

Checksum

- Checksum is a 16 bit field used for error control.
- It verifies the integrity of data in the TCP payload.

- Sender adds CRC checksum to the checksum field before sending the data.
- Receiver rejects the data that fails the CRC check.

Urgent Pointer

- Urgent pointer is a 16 bit field.
- It indicates how much data in the current segment counting from the first data byte is urgent.
- Urgent pointer added to the sequence number indicates the end of urgent data byte.
- This field is considered valid and evaluated only if the URG bit is set to 1.

Options

- Options field is used for several purposes.
- The size of options field varies from 0 bytes to 40 bytes.

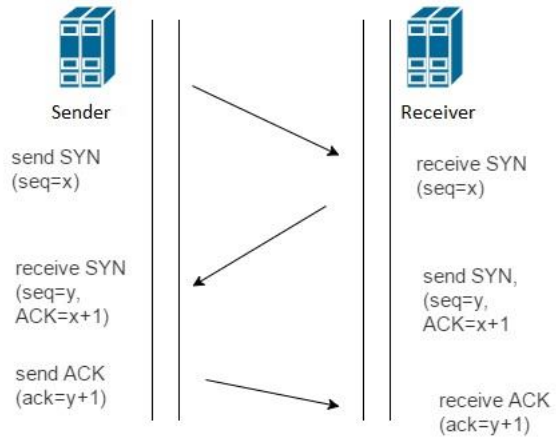
Options field is generally used for the following purposes-

A. Time stamp
B. Window size extension
C. Parameter negotiation
D. Padding

Connection Establishment in TCP

Since the connection establishment phase of TCP makes use of 3 packets, it is also known as three way handshaking **(SYN, SYN + ACK, ACK).**

- S**tep 1 (SYN) :** In the first step, client wants to establish a connection with server, so it sends a segment with SYN(Synchronize Sequence Number) which informs server that client is likely to start communication and with what sequence number it starts segments with.
- **Step 2 (SYN + ACK):** Server responds to the client request with SYN-ACK signal bits set. Acknowledgement (ACK) signifies the response of segment it received and SYN signifies with what sequence number it is likely to start the segments with.
- **Step 3 (ACK) :** In the final part client acknowledges the response of server and they both establish a reliable connection with which they will start the actual data transfer.



Connection Establishment in TCP

UDP vs TCP

- TCP proves to be an overhead for certain kinds of applications.
- The Connection Establishment Phase, Connection Termination Phase etc of TCP are time consuming.
- To avoid this overhead, certain applications which require fast speed and less overhead use UDP.

Congestion

Congestion is a state occurring in network layer when the message traffic is so heavy that it slows down network response time.

Reasons for Congestion

- Too many hosts in broadcast domain
- Low Bandwidth
- Outdated Hardware
- Bad network configuration

Effects of Congestion

- As delay increases, performance decreases.
- If delay increases, retransmission occurs, making situation worse.

How to reduce congestion ?

- Monitor Your Network Traffic.
- Segmentation of Network.
- Reconfigure TCP/IP Settings. .
- Choke Packet.
- Congestion Notification.

Congestion control algorithms

Leaky Bucket Algorithm

Imagine a bucket with a small hole in the bottom. No matter at what rate water enters the bucket, the outflow is at constant rate. When the bucket is full with water additional water entering spills over the sides and is lost.
Similarly, each network interface contains a leaky bucket and the following **steps** are involved in leaky bucket algorithm:

1. When host wants to send packet, packet is thrown into the bucket.
2. The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
3. Bursty traffic is converted to a uniform traffic by the leaky bucket.
4. In practice the bucket is a finite queue that outputs at a finite rate.

Token bucket Algorithm

The leaky bucket algorithm enforces output pattern at the average rate, no matter how bursty the traffic is. So in order to deal with the bursty traffic we need a flexible algorithm so that the data is not lost. One such algorithm is token bucket algorithm.

Steps of this algorithm can be described as follows:

1. In regular intervals tokens are thrown into the bucket.
2. The bucket has a maximum capacity.
3. If there is a ready packet, a token is removed from the bucket, and the packet is sent.
4. If there is no token in the bucket, the packet cannot be sent.

Token bucket vs Leaky bucket:

The leaky bucket algorithm controls the rate at which the packets are introduced in the network, but it is very conservative in nature. Some flexibility is introduced in the token bucket algorithm. In the token bucket, algorithm tokens are generated at each tick (up to a certain limit). For an incoming packet to be transmitted it must capture a token and the transmission takes place at the same rate. Hence some of the busty packets are transmitted at the same rate if tokens are available and thus introduces some amount of flexibility in the system.

Quality of Service

Quality of service (QoS) refers to any technology that manages data traffic to reduce packet loss, latency and jitter on the network. QoS controls and manages network resources by setting priorities for specific types of data on the network.

Networks need to provide predictable and measureable services as applications -- such as voice, video and delay-sensitive data -- traverse the network. Organizations use QoS to meet the traffic requirements of sensitive applications, such as real-time voice and video, and to prevent the degradation of quality caused by packet loss, delay and jitter.

Organizations can achieve QoS by using certain tools and techniques, such as jitter, buffer and traffic shaping. For many organizations, QoS is included in the service-level agreement with their network service provider to guarantee a certain level of performance.

QoS parameters

Organizations can measure QoS quantitatively by using several parameters, including the following:

- **Packet loss** happen when network links become congested and routers and switches start dropping packets. When packets are dropped during real-time communication, such as a voice or video calls, these sessions can experience jitter and gaps in speech.
- **Jitter** is the result of network congestion, timing drift and route changes. Too much jitter can degrade the quality of voice and video communication.
- **Latency** is the time it takes a packet to travel from its source to its destination. Latency should be as close to zero as possible. If a voice over IP call has a high amount of latency, it can experience echo and overlapping audio.
- **Bandwidth** is the capacity of a network communications link to transmit the maximum amount of data from one point to another in a given amount of time. QoS optimizes the network by managing bandwidth and setting priorities for applications that require more resources than others.
- **Mean opinion score** is a metric to rate voice quality that uses a five-point scale, with a five indicating the highest quality.

UNIT V

Application Layer

1.Introduction

2.Domain Name System (DNS)

3.World Wide Web (WWW)

4.Hypertext Transfer Protocol (HTTP)

5.TELNET

6.Simple Mail Transfer Protocol (SMTP)

7.File Transfer Protocol (FTP)

8.Simple Network Management Protocol (SNMP)

9.Introduction to Cryptography

10.Firewall

Application Layer

1. Introduction

Application layer is the top most layer of 7 layer OSI model and 4 layer TCP/IP protocol suite that provides the interface between the applications and network. It provides full end-user access to a variety of network services for efficient data flow.

The application layer provides many services such as

- **Network Virtual Terminal:** It allows a user to log on to a remote host. The application creates software emulation of a terminal at the remote host. User's computer talks to the software terminal which in turn talks to the host and vice versa. Then the remote host believes it is communicating with one of its own terminals and allows user to log on.
- **Directory Services:** This layer provides access for global information about various services.
- **Mail Services:** This layer provides the basis for E-mail forwarding and storage.
- **File Transfer, Access and Management (FTAM):** It is a standard mechanism to access files and manages it. Users can access files in a remote computer and manage it. They can also retrieve files from a remote computer.

The most widely used applications protocols are:

HTTP (Hypertext Transport Protocol)

The Hypertext Transfer Protocol delivers Web pages over the network.

Telnet

The Network Terminal Protocol, provides remote login over the network.

SMTP (Simple Mail Transfer Protocol)

The Simple Mail Transfer Protocol delivers electronic mail.

FTP (File Transfer Protocol)

The File Transfer Protocol is used for interactive file transfer.

Application Architecture

Application architecture is different from the network architecture. The network architecture is fixed and provides a set of services to applications. The application architecture, on the other hand, is designed by the application developer and defines how the application should be structured over the various end systems.

Application architecture is of two types

- **Client-server architecture**
- **P2P (peer-to-peer) architecture**

Client-server architecture: An application program running on the local machine sends a request to another application program is known as a client, and a program that serves a request is known as a server. For example, when a web server receives a request from the client host, it responds to the request to the client host. In Client-server architecture, clients do not directly communicate with each other. For example, in a web application, two browsers do not directly communicate with each other.

Client

A client is a program that runs on the local machine requesting service from the server. A client program is a finite program means that the service started by the user and terminates when the service is completed.

Server

A server is a program that runs on the remote machine providing services to the clients. When the client requests for a service, then the server opens the door for the incoming requests, but it never initiates the service.A server program is an infinite program means that when it starts, it runs infinitely unless the problem arises. The server waits for the incoming requests from the clients. When the request arrives at the server, then it responds to the request.

Disadvantage of Client-server architecture:

It is a single-server based architecture which is incapable of holding all the requests from the clients. For example, a social networking site can become overwhelmed when there is only one server exists.

P2P (peer-to-peer) architecture: It has no dedicated server. The peers are the computers which are not owned by the service provider. Most of the peers reside in the homes, offices, schools, and universities. The peers communicate with each other without passing the information through a dedicated server. This architecture is known as peer-to-peer architecture. The applications based on P2P architecture includes file sharing and internet telephony.

2. Domain Name System (DNS)

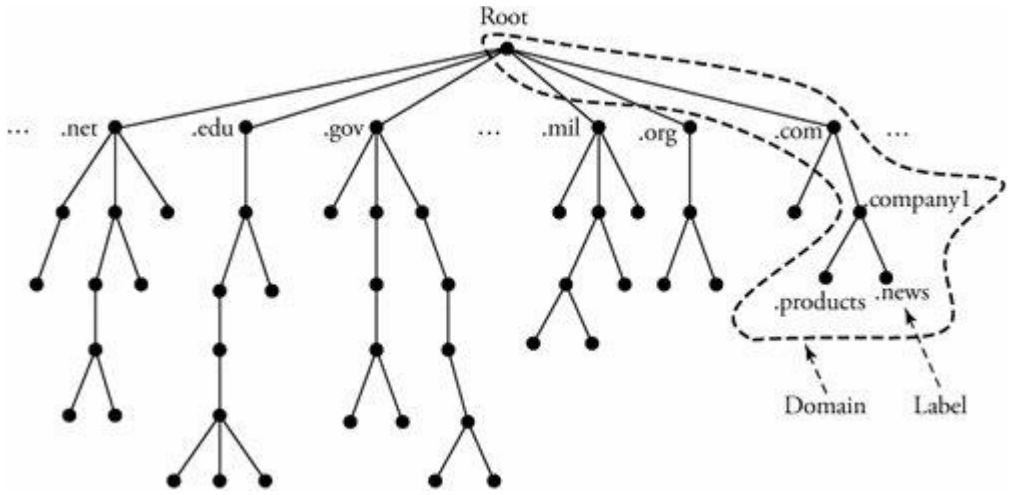
DNS is a protocol in the TCP/IP protocol suite. Its purpose is to convert easy-to-understand domain names like "**name.com**" into an Internet Protocol (IP) address, such as 10.20.134.42. An IP address is given to each device on the Internet, and that address is necessary to find the appropriate device. When a user wants to load a webpage, a translation must occur between what a user types into their web browser (name.com) and the IP address necessary to locate the name.com webpage.

DNS Architecture

The DNS comprises of Domains and Name Servers that have been described below

Domains

Domains are designed as a hierarchical structure in the Internet. This hierarchy has multiple levels from 0 to 127, with a root at the top. The top-level domains refer to the type of organization to which the network belongs, and sub domains further identify the specific network on which the host is situated. The following diagram shows the hierarchy:



In the above diagram each sub tree represents a domain. Each domain can be partitioned into sub domains and these can be further partitioned and so on.

While searching for a host, the DNS tree is traversed in an ascending order, starting from leaf nodes and moving towards the root. The dot (.), which is the root domain, is the starting point of the tree. In DNS, records are specified as the last character in the domain name. A domain is a portion in a domain name space. A Top-Level Domain (TLD) is a domain that directly branches off from the root of the tree. com, net, and org are some top-level domains.

Name Server

Name server contains the DNS database. This database comprises of various names and their corresponding IP addresses. Since it is not possible for a single server to maintain entire DNS database, therefore, the information is distributed among many DNS servers.

Types of Name Servers

-Root server
-Primary server
-Secondary server

DNS is a query/response protocol. The client queries information (for example the IP address corresponding to www.name.com). This request is followed by a reply from the DNS server.

Types of DNS queries:

Recursive query - In a recursive query, a DNS client requires that a DNS server (typically a DNS recursive resolver) will respond to the client with either the requested resource record or an error message if the resolver can't find the record.

Iterative query - in this situation the DNS client will allow a DNS server to return the best answer it can. If the queried DNS server does not have a match for the query name, it will return a referral to a DNS server authoritative for a lower level of the domain namespace. The DNS client will then make a query to the referral address. This process continues with additional DNS servers down the query chain until either an error or timeout occurs.

Non-recursive query - typically this will occur when a DNS resolver client queries a DNS server for a record that it has access to either because it's authoritative for the record or the record exists inside of its cache. Typically, a DNS server will cache DNS records to prevent additional bandwidth consumption and load on upstream servers.

When a user try to retrieve a web page from *name.com*, the following activities take place

- The query is received by a DNS recursive resolver which in turn queries a DNS root nameserver.
- The root server then responds to the resolver with the address of a Top Level Domain DNS server (.com), which stores the information for its domains.
- The resolver then makes a request to the .com Top Level Domain DNS server which responds with the IP address of the domain's nameserver.
- The recursive resolver sends a query to the domain's nameserver which will return the IP address for name.com to the resolver which will then respond to the web browser with the IP address of the domain requested initially.

Now the browser makes the request for the web page as follows

- The browser makes a HTTP request to the IP address.
- The server at that IP returns the webpage to be rendered in the browser.

3. World Wide Web (WWW)

World Wide Web (WWW), gives users access to a vast array of documents that are connected to each other by means of hypertext or hypermedia links—i.e., hyperlinks, electronic connections that link related pieces of information in order to allow a user easy access to them. Hypertext allows the user to select a word or phrase from text and thereby access other documents that contain additional information pertaining to that word or phrase. Hypermedia documents feature links to images, sounds, animations, and movies. Tim Berners-Lee developed the *World Wide Web* in 1989.

4. HyperText Transfer Protocol (HTTP)

The Hypertext Transfer Protocol (HTTP) is an application protocol that runs on top of the TCP/IP suite of protocols. It is a set of rules for transferring files, such as text, graphic images, sound, video, and other multimedia files, on the World Wide Web (WWW).

A Web server machine contains a program that is designed to wait for HTTP requests and handle them when they arrive. A Web browser is an HTTP client, sending requests to server machines. When the browser user enters web page requests by typing a Uniform Resource Locator (URL) or clicking on a hypertext link the browser builds an HTTP request and sends it to the IP address indicated by the URL. The destination server machine receives the request and sends back the requested page.

HTTPS

HTTPS - HTTP over SSL or HTTP Secure is the use of Secure Sockets Layer (SSL) or Transport Layer Security (TLS) as a sublayer under regular HTTP application layering. The problem with the regular HTTP protocol is that the information that flows from server to browser is not encrypted, which means it can be easily stolen. HTTPS encrypts and decrypts user HTTP page requests as well as the pages that are returned by the Web server. The use of HTTPS protects against eavesdropping. HTTPS was developed by Netscape.

5. TELNET

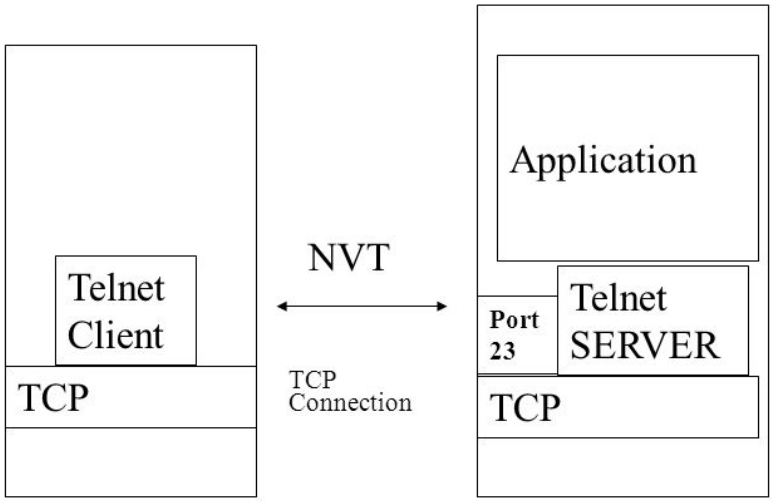
The TELNET protocol enables a user to connect and log on to any other hosts in the network from user's own computer by offering a remote log on capability. It is widely used as a terminal emulator.

Connection establishment using TELNET

To establish connections, TELNET uses the TCP protocol. The TELNET service is offered in the host machine's TCP port 23. The user at the terminal interacts with the local telnet client. The TELNET client acts as a terminal accepting any keystrokes from the keyboard, interpreting them and displaying the output on the screen. The client on the computer makes the TCP connection to the host machine's port 23 where the TELNET server answers. The TELNET server interacts with applications in the host machine and assists in the terminal emulation.

As the connection is setup, the both ends of the TELNET connection are assumed to be originated and terminated at the **Network Virtual Terminal** (NVT). The NVT is a network wide terminal which is host independent so that both the server and the client in the connection may not need to keep any information about each other terminal's characteristics as both sees each other as a NVT terminal. As there are several types of terminals, which may be able to provide additional services from those provided by the NVT, the TELNET protocol contains a negotiation method for the user and the server to negotiate changes to the terminal provided in the NVT.

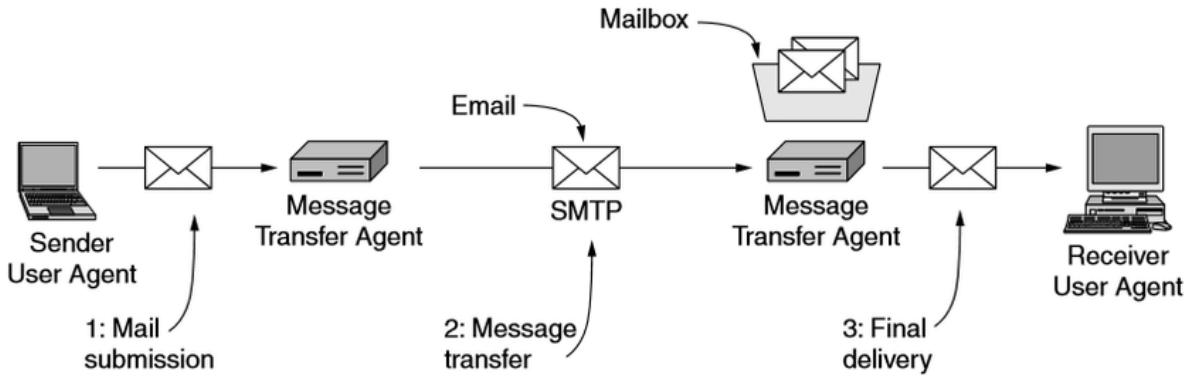
TELNET ARCHITECTURE



6. Simple Mail Transfer Protocol (SMTP)

SMTP is an application layer protocol. It allows software to transmit an electronic mail over the internet. SMTP is a protocol is used to send the mail. POP (Post Office Protocol) or IMAP (Internet Message Access Protocol) are used to retrieve those mails at the receiver's side. The client who wants to send the mail opens a TCP connection to the SMTP server and then sends the mail across the connection. The SMTP server is always on listening mode. As soon as it listens for a TCP connection from any client, the SMTP process initiates a connection on port 25. After successfully establishing the TCP connection, the client process sends the mail instantly. The main purpose of SMTP is to set up communication rules between servers. The servers have a way of identifying themselves and announcing what kind of communication they are trying to perform. They also have a way of handling the errors such as incorrect email address.

Two components namely User Agent (UA) and Mail Transfer Agent (MTA) are involved in the mail transfer. The user agent (UA) prepares the message, creates the envelope and then puts the message in the envelope. The mail transfer agent (MTA) transfers this mail across the internet.



Working of SMTP

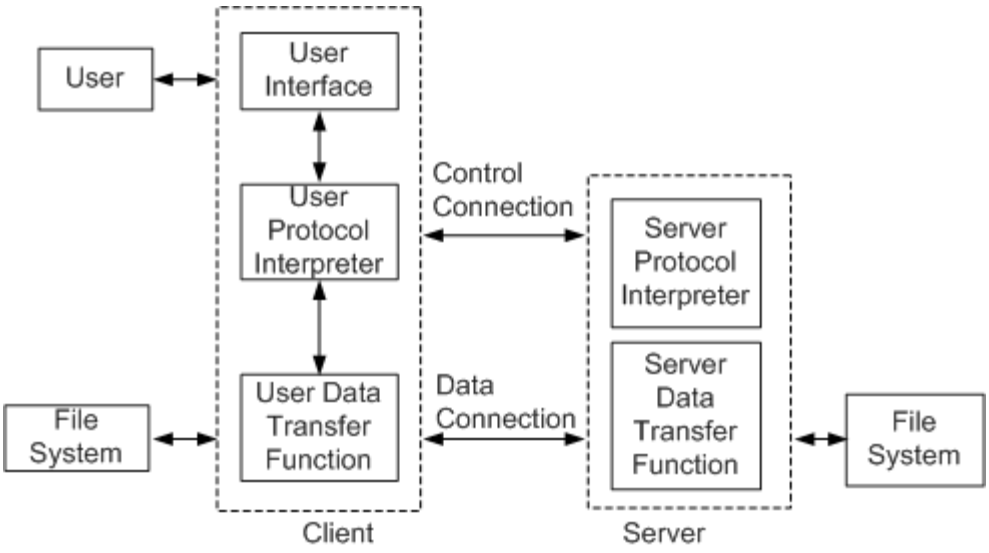
- **Composition of Mail:** A user sends an e-mail by composing an electronic mail message using a Mail User Agent (MUA). Mail User Agent is a program which is used to send and receive mail. The message contains two parts: body and header. The body is the main part of the message while the header includes information such as the sender and recipient address. The header also includes descriptive information such as the subject of the message. In this case, the message body is like a letter and header is like an envelope that contains the recipient's address.
- **Submission of Mail:** After composing an email, the mail client then submits the completed e-mail to the SMTP server by using SMTP on TCP port 25.
- **Delivery of Mail:** E-mail addresses contain two parts: username of the recipient and domain name, for example ravi@abcd.com.If the domain name of the recipient's email address is different from the sender's domain name, then MSA will send the mail to the Mail Transfer Agent (MTA). To relay the email, the MTA will find the target domain. It checks the MX record from Domain Name System to obtain the target domain. The MX record contains the domain name and IP address of the recipient's domain. Once the record is located, MTA connects to the exchange server to relay the message.
- **Receipt and Processing of Mail:** Once the incoming message is received, the exchange server delivers it to the incoming server (Mail Delivery Agent) which stores the e-mail where it waits for the user to retrieve it.
- **Access and Retrieval of Mail:** The stored email in MDA can be retrieved by using MUA (Mail User Agent). MUA can be accessed by using login and password.

7. File Transfer Protocol (FTP)

The File Transfer Protocol (FTP) is used to copy files between two computer systems over the TCP connection. The FTP overcomes the problem of different file systems used in the network.

In the FTP, the user communicates with a user interface in the local FTP client process. The local FTP client process makes a control connection to the remote server's FTP server protocol. FTP server protocol is located in the TCP port 21. The local FTP client acts as a protocol interpreter who interprets the user commands to the acronyms used between the client and the server protocol. The control connection is basically a simple TELNET's NVT session. The client sends commands across the control connection to the server. The server replies to the messages according to the server protocol.

When the user request a data transfer, a special data connection is opened between the server and the client and the files are sent through this connection. Separate data transfer process created for the server and the client. The data connection exists until the command that it was created for is executed.



FTP components

Trivial File Transfer Protocol (TFTP)

Trivial File Transfer Protocol (**TFTP**) is an Internet protocol for transferring files that is simpler to use than the File Transfer Protocol (FTP) but less capable. It is used where user authentication and directory visibility are not required. Due to simplicity, the TFTP is mostly used for loading operating system for the diskless workstations or download the initialization and configuration files for the software during the boot phase. The TFTP transfers its data as UDP datagrams.

The TFTP sends data as 512 bytes blocks including 4 byte header. Each block is numbered in the header. The numbering starts from 1. Either ASCII or binary information can be transferred. The TFTP is capable of sending and receiving files. The receiving file is done with the read request primitive and the sending with the write request primitive. The client gets a free port and sends a read or write request primitive to the server's UDP port 69. The server changes the port to other, which it will use for the rest of the session while transferring the files with the client. Since the all the blocks should be the size of 512 bytes, the end of file block is indicated with a block, which size is less than 512 bytes. The read request is replied with the DATA datagram from the server. The client acknowledges the DATA and the server sends DATA again and the client acknowledges them as well. This is repeated until the end of file is reached or an error is found. The write request on the other hand is acknowledged by the server and the client can then start sending data. The server acknowledges the data until the end of file is reached or error is found.

8. Simple Network Management Protocol (SNMP)

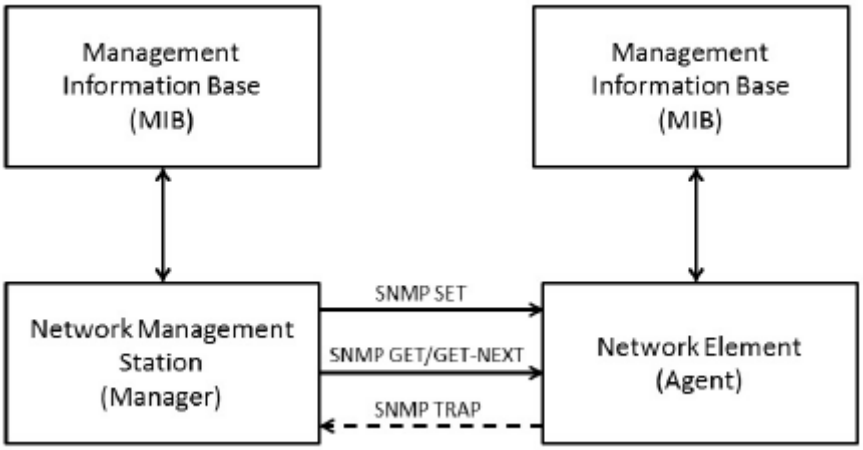
SNMP is a standard protocol for managing devices in an Internet using the TCP/IP protocol suite. It is an open technology that enables the management of networks and devices, or nodes, connected to the network. SNMP messages are sent over IP using the User Datagram Protocol (UDP). SNMP is an application-level protocol. The protocol can monitor devices made by different manufacturers and installed on different networks.

Network Management Architecture

Network management system contains a manager and agents. The Manager is the console through which the network administrator performs network management functions. Agents are the entities that interface to the actual device being managed. Bridges, Hubs, Routers or

network servers are examples of managed devices that contain managed objects. These managed objects might be hardware, configuration parameters, performance statistics, and so on, that directly relate to the current operation of the device in question. These objects are arranged in a virtual information database, called a Management Information Base (MIB). SNMP allows managers and agents to communicate for the purpose of accessing these objects.

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



Architecture of SNMP

The SNMP agent receives requests on UDP port 161. The manager may send requests from any available source port to port 161 in the agent. The agent response will be sent back to the source port on the manager. The manager receives notifications on port 162. The agent may generate notifications from any available port.

SNMP has specific roles in network management. It defines the format of the packet to be sent from a manager to an agent and vice versa. It also interprets the result and creates statistics (often with the help of other management software). It uses Structure of Management Information (SMI) and Management Information Base (MIB). SMI defines the

general rules for naming objects, defining object types (including range and length), and showing how to encode objects and values. SMI does not define the number of objects an entity should manage or name the objects to be managed. SMI is a guideline for SNMP. For each entity to be managed, MIB must define the number of objects, name them according to therules defined by SMI, and associate a type to each named object.

9. Introduction to Cryptography

Cryptography is the study and practice of techniques for secure communication in the presence of third parties called adversaries. It deals with developing and analyzing protocols which prevents malicious third parties from retrieving information being shared between two entities thereby following the various aspects of information security.

Secure Communication refers to the scenario where the message or data shared between two parties can't be accessed by an adversary. In Cryptography, an Adversary is a malicious entity, which aims to retrieve precious information or data thereby undermining the principles of information security.

Cryptographic techniques provide the following security services

- **Confidentiality** refers to certain rules and guidelines usually executed under confidentiality agreements which ensure that the information is restricted to certain people or places.
- **Integrity** refers to maintaining and making sure that the data stays accurate and consistent over its entire life cycle.
- **Authentication** is the process of making sure that the piece of data being claimed by the user belongs to it.
- **Non-repudiation** refers to ability to make sure that a person or a party associated with a contract or a communication cannot deny the authenticity of their signature over their document or the sending of a message.

Consider two end users Sam and Raj. Now, Sam wants to send a message M to Raj in a secured manner. Then the sender's message (called the Plaintext), is converted into an unreadable form using a Key k. The resultant text obtained is called the Ciphertext. This process is known as **Encryption**. At the receiving end, the Ciphertext is converted back into the plaintext using the same Key k, so that it can be read by the receiver. This process is known as **Decryption**.

10.Firewall

Firewall is a tool that can be used to enhance the security of computers connected to a network, such as a LAN or the Internet. A firewall separates a computer from the Internet, inspecting packets of data as they arrive at either side of the firewall to determine whether it should be allowed to pass or be blocked.

.

Firewalls act as guards at the computer's entry points where the computer exchanges data with other devices on the network. Firewalls ensure that packets that are requesting permission to enter the computer meet certain rules that are established by the user of the computer. Firewalls operate in two ways, by either denying or accepting all messages based on a list of designated acceptable or unacceptable sources or ports.

Types of Firewalls

Packet Fitering

The most common firewall method is known as packet filtering. A packet filter firewall receives a packet from the Internet, it checks information held in the IP Address in the header of the packet and checks it against a table of access control rules to determine whether or not the packet is acceptable. In this case, a set of rules established by the firewall administrator serves the purpose. These rules may specify certain actions when a particular source or destination IP address or port number is identified. For example, access to a web site can be blocked by designating the IP address of that site as a non-permitted connection (incoming or outgoing) with the user's computer. When the packet filter firewall encounters a packet from that site, it examines the packet. Since IP address of that site is contained in the header of the packet, it meets the conditions that specifically deny such a connection and the web traffic is not permitted to go through.

Drawbacks

One method of getting around a packet filter firewall is known as IP spoofing, in which hackers adopt the IP address of a trusted source, thereby fooling the firewall into thinking that the packets from the hacker are actually from a trusted source. The second problem with packet filter firewalls is, once an initial connection has been approved by the firewall, the source computer is connected directly to the destination computer, thereby potentially exposing the destination computer and all the computers to which it is connected to attack.

Stateful Packet Inspection

A second method utilized by firewalls is known as stateful packet inspection. It examines not just the headers of the packet, but also the contents, to determine more about the packet than just its source and destination information. It is called a stateful packet inspection because it examines the contents of the packet to determine the state of the communication i.e. it ensures that the stated destination computer has previously requested the current communication. This is a way of ensuring that all communications are initiated by the recipient computer and are taking place only with sources that are known and trusted from previous interactions. In addition to being more rigorous in their inspection of packets, stateful inspection firewalls also close off ports until connection to the specific port is requested. This allows an added layer of protection from the threat of port scanning.

Application Level Proxy

An application-level proxy determines whether a connection to a requested application is permitted. Only connections for specified purposes, such as Internet access or e-mail, will be permitted. This allows system administrators to control what applications their organisation's computers will be used for.