Internet Protocols

Transmission Control Protocol (TCP)

TCP is a connection oriented protocol and offers end-to-end packet delivery. It acts as back bone for connection.It exhibits the following key features:

- Transmission Control Protocol (TCP) corresponds to the Transport Layer of OSI Model.

- TCP is a reliable and connection oriented protocol.

- TCP offers:

  - Stream Data Transfer.

  - Reliability.

  - Efficient Flow Control

  - Full-duplex operation.

  - Multiplexing.

- TCP offers connection oriented end-to-end packet delivery.

- TCP ensures reliability by sequencing bytes with a forwarding acknowledgement number that indicates to the destination the next byte the source expect to receive.

- It retransmits the bytes not acknowledged with in specified time period.

TCP Services

TCP offers following services to the processes at the application layer:

- Stream Delivery Service

- Sending and Receiving Buffers

- Bytes and Segments

- Full Duplex Service

- Connection Oriented Service

- Reliable Service

Stream Deliver Service

TCP protocol is stream oriented because it allows the sending process to send data as stream of bytes and the receiving process to obtain data as stream of bytes.

Sending and Receiving Buffers

It may not be possible for sending and receiving process to produce and obtain data at same speed, therefore, TCP needs buffers for storage at sending and receiving ends.

Bytes and Segments

The Transmission Control Protocol (TCP), at transport layer groups the bytes into a packet. This packet is called segment. Before transmission of these packets, these segments are encapsulated into an IP datagram.

Full Duplex Service

Transmitting the data in duplex mode means flow of data in both the directions at the same time.

Connection Oriented Service

TCP offers connection oriented service in the following manner:

1. TCP of process-1 informs TCP of process – 2 and gets its approval.

2. TCP of process – 1 and TCP of process – 2 and exchange data in both the two directions.

3. After completing the data exchange, when buffers on both sides are empty, the two TCP's destroy their buffers.

Reliable Service

For sake of reliability, TCP uses acknowledgement mechanism.

Internet Protocol (IP)

Internet Protocol is **connectionless** and **unreliable** protocol. It ensures no guarantee of successfully transmission of data.

In order to make it reliable, it must be paired with reliable protocol such as TCP at the transport layer.

Internet protocol transmits the data in form of a datagram as shown in the following diagram:
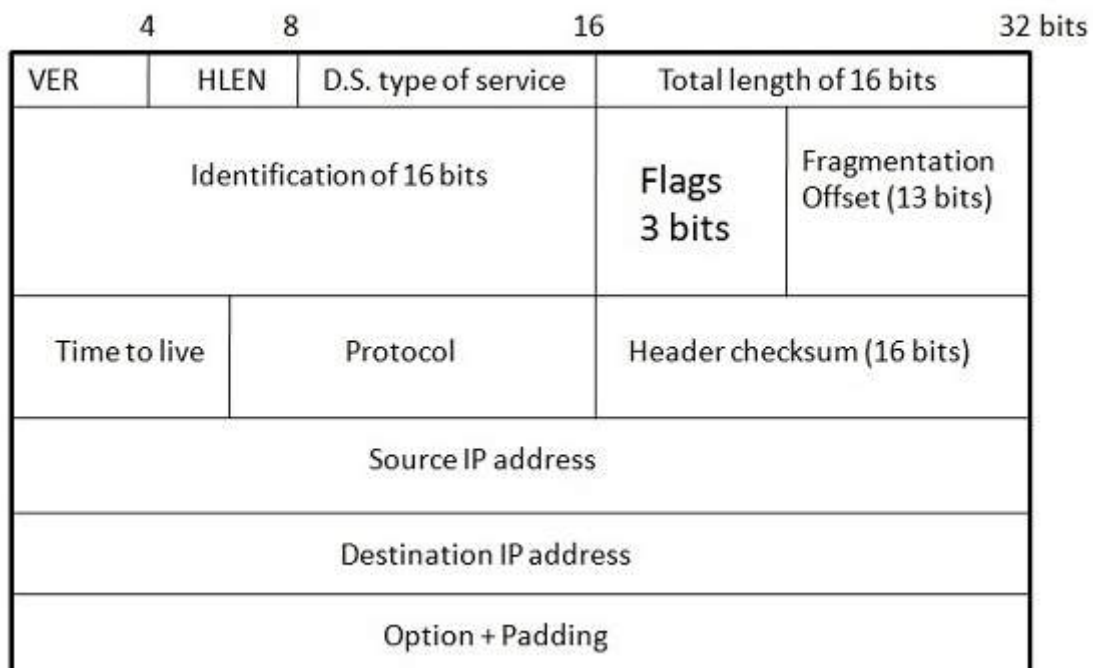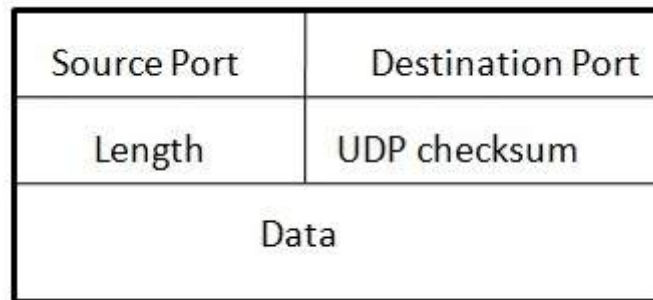

Points to remember:

- The length of datagram is variable.

- The Datagram is divided into two parts: **header** and **data.**

- The length of header is 20 to 60 bytes.

- The header contains information for routing and delivery of the packet.

User Datagram Protocol (UDP)

Like IP, UDP is connectionless and unreliable protocol. It doesn't require making a connection with the host to exchange data. Since UDP is unreliable protocol, there is no mechanism for ensuring that data sent is received.

UDP transmits the data in form of a datagram. The UDP datagram consists of five parts as shown in the following diagram:

Source Port	Destination Port
Length	UDP checksum
Data	

Points to remember:

- UDP is used by the application that typically transmit small amount of data at one time.

- UDP provides protocol port used i.e. UDP message contains both source and destination port number, that makes it possible for UDP software at the destination to deliver the message to correct application program.

File Transfer Protocol (FTP)

FTP is used to copy files from one host to another. FTP offers the mechanism for the same in following manner:

- FTP creates two processes such as Control Process and Data Transfer Process at both ends i.e. at client as well as at server.

- FTP establishes two different connections: one is for data transfer and other is for control information.

- **Control connection** is made between **control processes** while **Data Connection** is made between

- FTP uses **port 21** for the control connection and **Port 20** for the data connection.
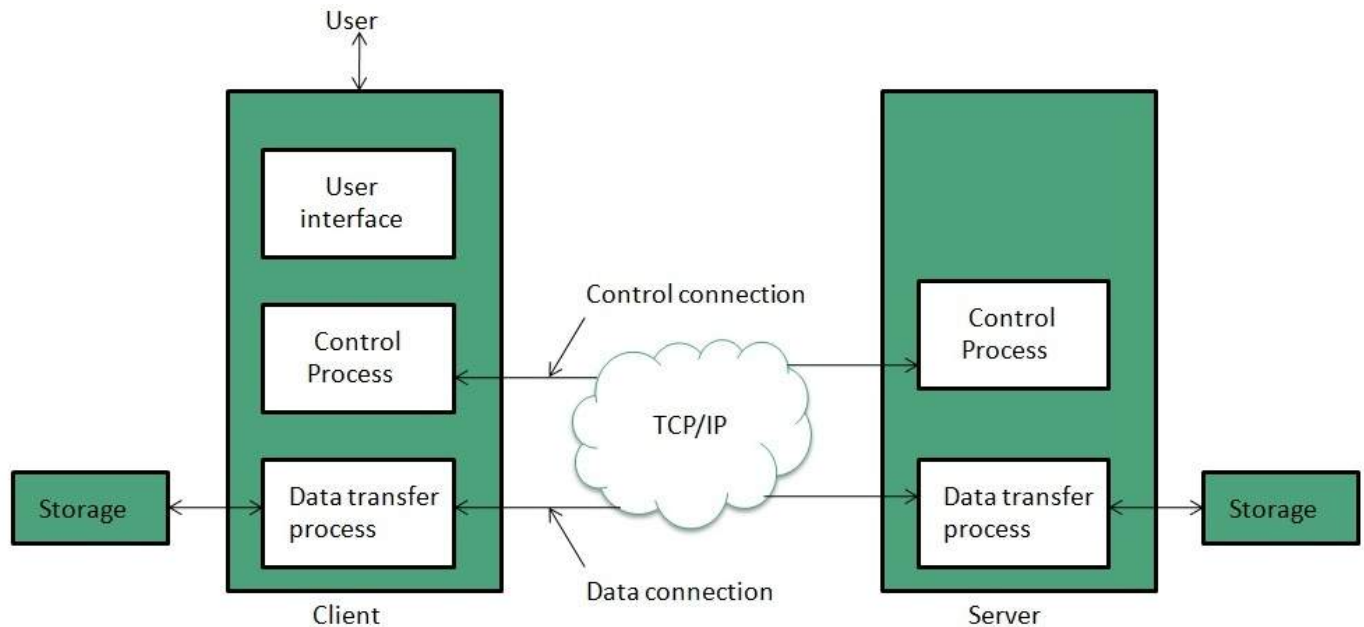
Trivial File Transfer Protocol (TFTP)

Trivial File Transfer Protocol is also used to transfer the files but it transfers the files without authentication. Unlike FTP, TFTP does not separate control and data information. Since there is no authentication exists, TFTP lacks in security features therefore it is not recommended to use TFTP.

Key points

- TFTP makes use of UDP for data transport. Each TFTP message is carried in separate UDP datagram.

- The first two bytes of a TFTP message specify the type of message.

- The TFTP session is initiated when a TFTP client sends a request to upload or download a file.

- The request is sent from an ephemeral UDP port to the **UDP port 69** of an TFTP server.

Difference between FTP and TFTP

S.N.	Parameter	FTP	TFTP
1	Operation	Transferring Files	Transferring Files
2	Authentication	Yes	No
3	Protocol	TCP	UDP
4	Ports	21 – Control, 20 – Data	Port 3214, 69, 4012
5	Control and Data	Separated	Separated
6	Data Transfer	Reliable	Unreliable

Telnet

Telnet is a protocol used to log in to remote computer on the internet. There are a number of Telnet clients having user friendly user interface. The following diagram shows a person is logged in to computer A, and from there, he remote logged into computer B.
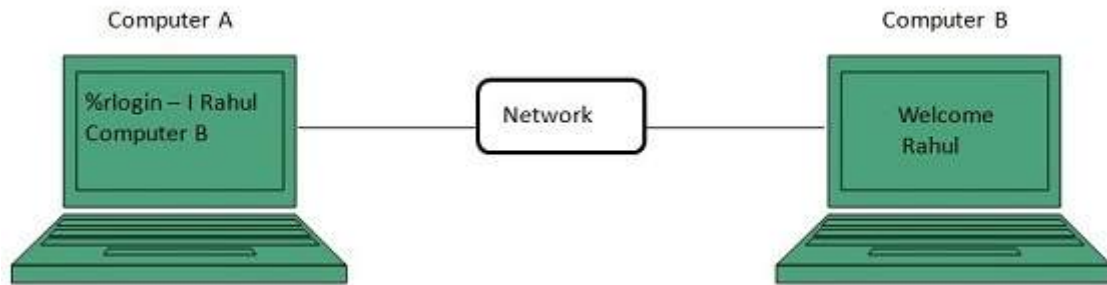
Hyper Text Transfer Protocol (HTTP)

HTTP is a communication protocol. It defines mechanism for communication between browser and the web server. It is also called request and response protocol because the communication between browser and server takes place in request and response pairs.

HTTP Request

HTTP request comprises of lines which contains:

- Request line
- Header Fields
- Message body

Key Points

- The first line i.e. the **Request line** specifies the request method i.e. **Get** or **Post.**
- The second line specifies the header which indicates the domain name of the server from where index.htm is retrieved.

HTTP Response

Like HTTP request, HTTP response also has certain structure. HTTP response contains:

- Status line
- Headers
- Message body