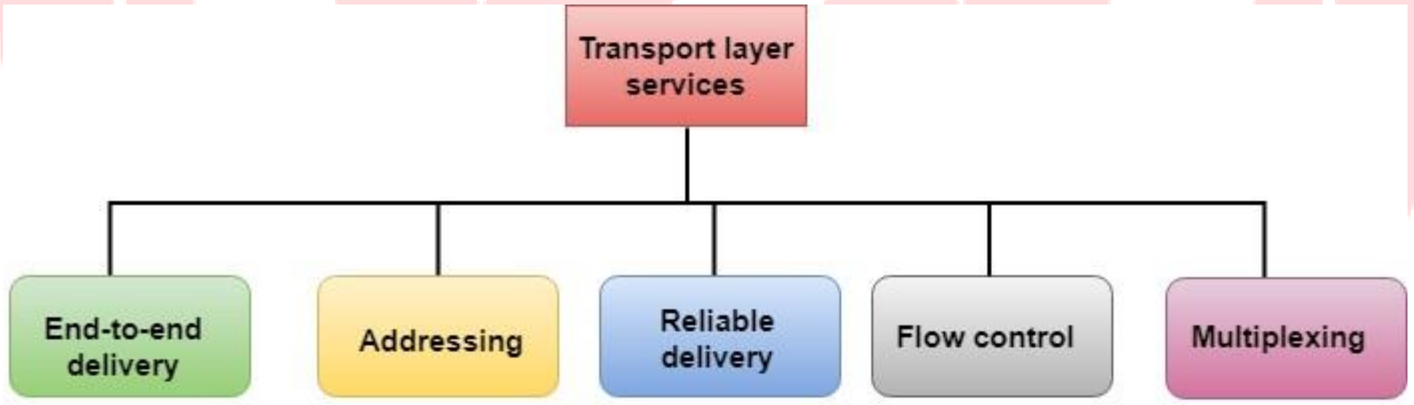
Unit-4 : Transport Layer Protocols

❖ Services provided by the Transport Layer

The services provided by the transport layer are similar to those of the data link layer. The data link layer provides the services within a single network while the transport layerprovides the services across an internetwork made up of many networks. The data link layer controls the physical layer while the transport layer controls all the lower layers.

The services provided by the transport layer protocols can be divided into five categories:

- End-to-end delivery
- Addressing
- Reliable delivery
- Flow control
- Multiplexing



❖ End-to-end delivery:

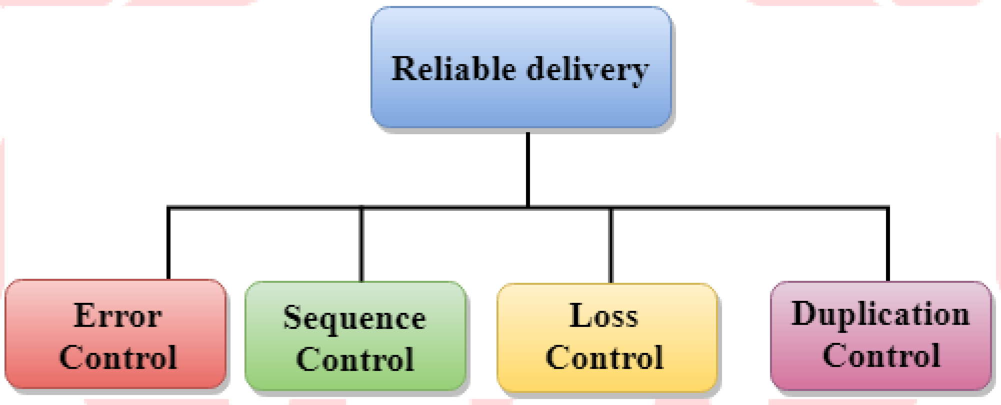
The transport layer transmits the entire message to the destination. Therefore, it ensures the end-to-end delivery of an entire message from a source to the destination.

❖ Reliable delivery:

The transport layer provides reliability services by retransmitting the lost and damaged packets.

The reliable delivery has four aspects:

- Error control
- Sequence control
- Loss control
- Duplication control



Error Control

- The primary role of reliability is **Error Control**. In reality, no transmission will be 100 percent error-free delivery. Therefore, transport layer protocols are designed to provide error-free transmission.

Sequence Control

- The second aspect of the reliability is sequence control which is implemented at the transport layer.

Loss Control

- Loss Control is a third aspect of reliability. The transport layer ensures that all the fragments of a transmission arrive at the destination, not some of them..

Duplication Control

- Duplication Control is the fourth aspect of reliability. The transport layer guarantees that no duplicate data arrive at the destination.

❖ Flow Control

Flow control is used to prevent the sender from overwhelming the receiver.

If the receiver is overloaded with too much data, then the receiver discards the packets and asking for the retransmission of packets. This increases network congestion and thus, reducing the system performance. The transport layer is responsible for flow control.

❖ Multiplexing

he transport layer uses the multiplexing to improve transmission efficiency.

Multiplexing can occur in two ways:

- **Upward multiplexing:** Upward multiplexing means multiple transport layer connections use the same network connection. To make more cost-effective, the transport layer sends several transmissions bound for the same destination along the same path; this is achieved through upward multiplexing.
- **Downward multiplexing:** Downward multiplexing means one transport layer connection uses the multiple network connections. Downward multiplexing allows the transport layer to split a connection among several paths to improve the throughput. This type of multiplexing is used when networks have a low or slow capacity.

❖ Addressing

- According to the layered model, the transport layer interacts with the functions of the session layer. Many protocols combine session, presentation, and application layer protocols into a single layer known as the application layer. In these cases, delivery to the session layer means the delivery to the application layer. Data generated by an application on one machine must be transmitted to the correct application on another machine. In this case, addressing is provided by the transport layer.
- The transport layer provides the user address which is specified as a station or port. The port variable represents a particular TS user of a specified station known as a Transport Service access point (TSAP). Each station has only one transport entity.

❖ Port Number

A port number is a unique identifier used with an IP address. A port is a 16-bit unsigned integer, and the total number of ports available in the TCP/IP model is 65,535 ports. Therefore, the range of port numbers is 0 to 65535. In the case of TCP, the zero-port number is reserved and cannot be used, whereas, in UDP, the zero port is not available. IANA (Internet Assigned Numbers Authority) is a standard body that assigns the port numbers.

Example of port number:

192.168.1.100: 7

In the above case, **192.168.1.100** is an IP address, and **7** is a port number.

To access a particular service, the port number is used with an IP address. The range from 0 to 1023 port numbers are reserved for the standard protocols, and the other port numbers are user-defined.

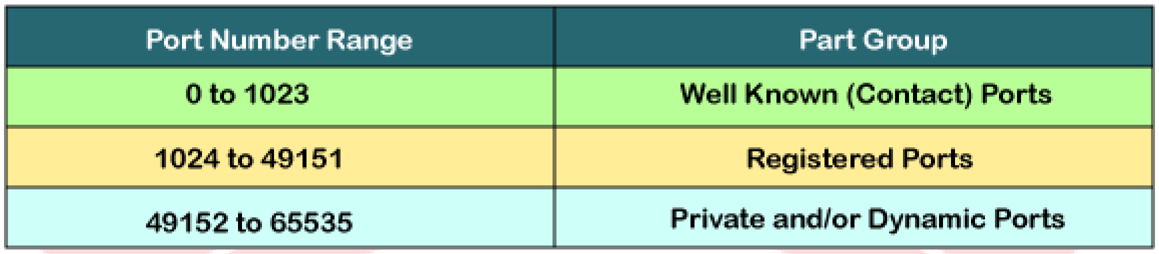
Why do we require port numbers?

A single client can have multiple connections with the same server or multiple servers. The client may be running multiple applications at the same time. When the client tries to access some service, then the IP address is not sufficient to access the service. To access the service from a server, the port number is required. So, the transport layer plays a major role in providing multiple communication between these applications by assigning a port number to the applications.

Classification of port numbers

The port numbers are divided into three categories:

- Well-known ports
- Registered ports
- Dynamic ports



Port Number Range	Part Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

Well-known ports

The range of well-known port is 0 to 1023. The well-known ports are used with those protocols that serve common applications and services such as HTTP (hypertext transfer protocol), IMAP (Internet Message Access Protocol), SMTP (Simple Mail Transfer Protocol), etc. For example, we want to visit some websites on an internet; then, we use http protocol; the http is available with a port number 80, which means that when we use

http protocol with an application then it gets port number 80. It is defined that whenever http protocol is used, then port number 80 will be used. Similarly, with other protocols such as SMTP, IMAP; well-known ports are defined. The remaining port numbers are used for random applications.

Registered ports

The range of registered port is 1024 to 49151. The registered ports are used for the user processes. These processes are individual applications rather than the common applications that have a well-known port.

Dynamic ports

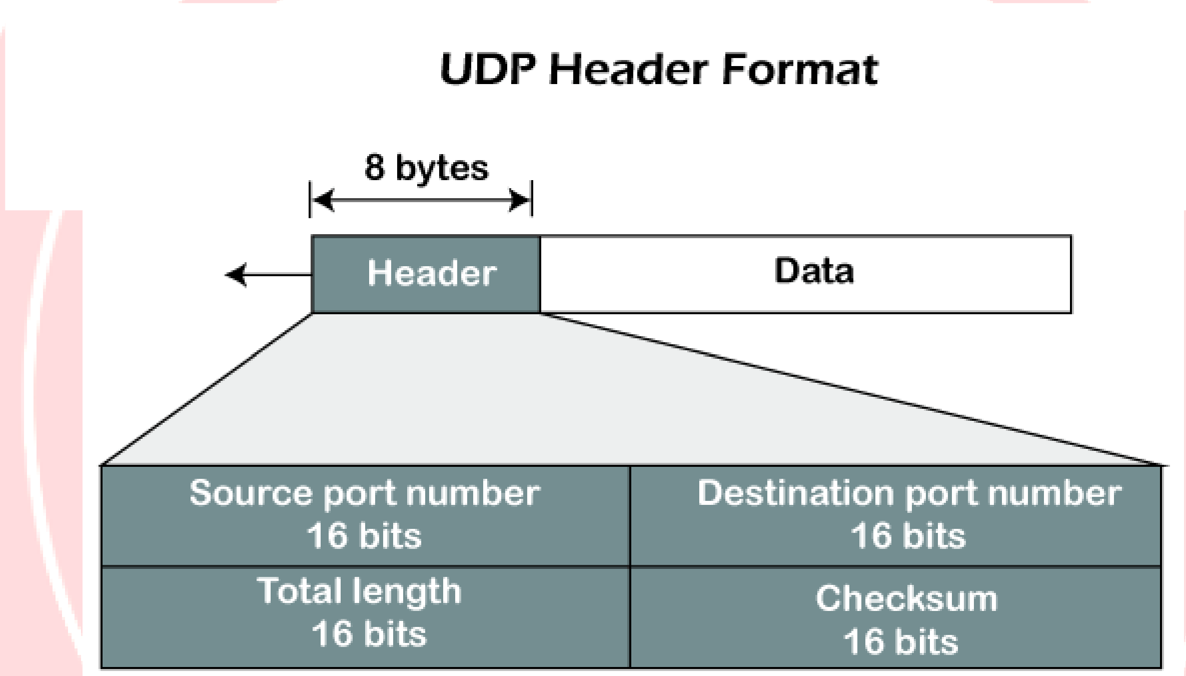
The range of dynamic port is 49152 to 65535. Another name of the dynamic port is ephemeral ports. These port numbers are assigned to the client application dynamically whena client creates a connection. The dynamic port is identified when the client initiates the connection, whereas the client knows the well-known port prior to the connection. This port is not known to the client when the client connects to the service.

❖ UDP Protocol

- UDP stands for **User Datagram Protocol**.
- UDP is a simple protocol and it provides nonsequenced transport functionality.
- UDP is a connectionless protocol.

- This type of protocol is used when reliability and security are less important than speed and size.
- UDP is an end-to-end transport level protocol that adds transport-level addresses, checksum error control, and length information to the data from the upper layer.
- The packet produced by the UDP protocol is known as a user datagram.

❖ UDP Header Format



In UDP, the header size is 8 bytes, and the packet size is upto 65,535 bytes. But this packet size is not possible as the data needs to be encapsulated in the IP datagram, and an IP packet,the header size can be 20 bytes; therefore, the maximum of UDP would be 65,535 minus 20. The size of the data that the UDP packet can carry would be 65,535 minus 28 as 8 bytes for the header of the UDP packet and 20 bytes for IP header.

The UDP header contains four fields:

- **Source port number:** It is 16-bit information that identifies which port is going t send the packet.
- **Destination port number:** It identifies which port is going to accept the information. It is 16-bit information which is used to identify application-level service on the destination machine.

- **Length:** It is 16-bit field that specifies the entire length of the UDP packet that includes the header also. The minimum value would be 8-byte as the size of the header is 8 bytes.
- **Checksum:** It is a 16-bits field, and it is an optional field. This checksum field checks whether the information is accurate or not as there is the possibility that the information can be corrupted while transmission.

❖ Services of UDP protocol

- **Transport layer protocol**

UDP is the simplest transport layer communication protocol. It contains a minimum amount of communication mechanisms. It is considered an unreliable protocol, and it is based on best- effort delivery services. UDP provides no acknowledgment mechanism, which means that the receiver does not send the acknowledgment for the received packet, and the sender also does notwait for the acknowledgment for the packet that it has sent.

- **Connectionless**

The UDP is a connectionless protocol as it does not create a virtual path to transfer the data. It does not use the virtual path, so packets are sent in different paths between the sender and the receiver, which leads to the loss of packets or received out of order.

- **Ordered delivery of data is not guaranteed.**

In the case of UDP, the datagrams are sent in some order will be received in the same order is not guaranteed as the datagrams are not numbered.

- **Ports**

The UDP protocol uses different port numbers so that the data can be sent to the correct destination. The port numbers are defined between 0 and 1023.

- **Faster transmission**

UDP enables faster transmission as it is a connectionless protocol, i.e., no virtual path is required to transfer the data. But there is a chance that the individual packet is lost, which affects the transmission quality. On the other hand, if the packet is lost in TCP connection, that packet will be resent, so it guarantees the delivery of the data packets.

- **Acknowledgment mechanism**

The UDP does have any acknowledgment mechanism, i.e., there is no handshaking between the UDP sender and UDP receiver. If the message is sent in TCP, then the receiver acknowledges that I am ready, then the sender sends the data. In the case of TCP, the handshaking occurs between the sender and the receiver, whereas in UDP, there is no handshaking between the sender and the receiver.

- **Segments are handled independently.**

Each UDP segment is handled individually of others as each segment takes different path to reach the destination. The UDP segments can be lost or delivered out of order to reach the destination as there is no connection setup between the sender and the receiver.

- **Stateless**

It is a stateless protocol that means that the sender does not get the acknowledgement for the packet which has been sent.

❖ UDP application

Here are few applications where UDP is used to transmit data:

- Domain Name Services
- Simple Network Management Protocol
- Trivial File Transfer Protocol
- Routing Information Protocol
- Kerberos

❖

❖ TCP

- TCP stands for Transmission Control Protocol.
- It provides full transport layer services to applications.
- It is a connection-oriented protocol means the connection established between both the ends of the transmission. For creating the connection, TCP generates a virtual circuit between sender and receiver for the duration of a transmission.

❖ TCP Services

- Stream Delivery Service.
- Sending and Receiving Buffers.
- Bytes and Segments.
- Full Duplex Service
- Connection Oriented Service.
- Reliable Service.

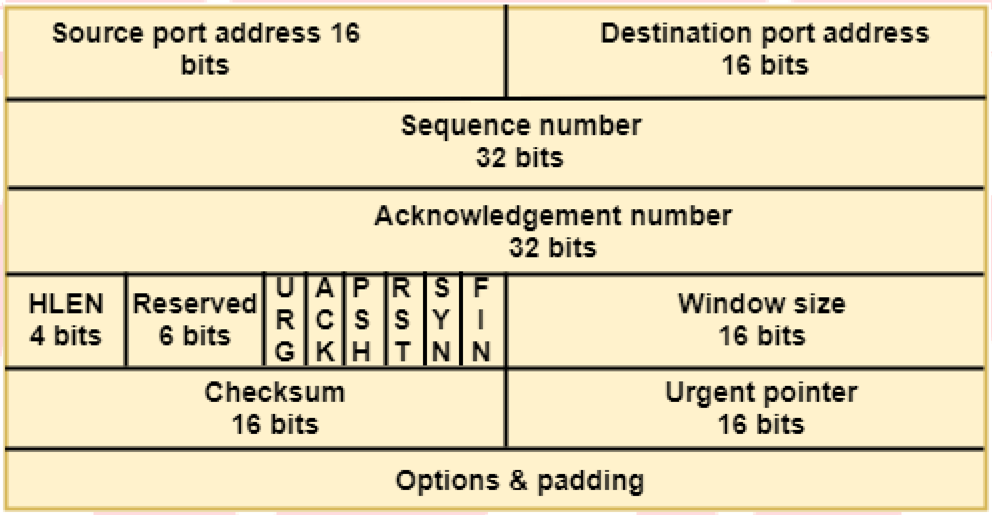
❖ TCP Features

- **Stream data transfer:** TCP protocol transfers the data in the form of contiguous stream of bytes. TCP group the bytes in the form of TCP segments and then passed it to the IP layer for transmission to the destination. TCP itself segments the data and forward to the IP.
- **Reliability:** TCP assigns a sequence number to each byte transmitted and expects a positive acknowledgement from the receiving TCP. If ACK is not received within a timeout interval, then the data is retransmitted to the destination.
The receiving TCP uses the sequence number to reassemble the segments if they arrive out of order or to eliminate the duplicate segments.
- **Flow Control:** When receiving TCP sends an acknowledgement back to the sender indicating the number the bytes it can receive without overflowing its internal buffer. The number of bytes is sent in ACK in the form of the highest sequence number that it can receive without any problem. This mechanism is also referred to as a window mechanism.
- **Multiplexing:** Multiplexing is a process of accepting the data from different applications and forwarding to the different applications on different computers. At the receiving end, the data is forwarded to the correct application. This process is known as demultiplexing. TCP transmits the packet to the correct application by using the logical channels known as ports.
- **Logical Connections:** The combination of sockets, sequence numbers, and window sizes, is called a logical connection. Each connection is identified by the pair of sockets used by sending and receiving processes.
- **Full Duplex:** TCP provides Full Duplex service, i.e., the data flow in both the directions at the same time. To achieve Full Duplex service, each TCP should have sending and receiving buffers so that the segments can flow in both the directions. TCP is a connection-oriented

V2V EDTECH LLP
DIPLOMA | DEGREE | BSCIT/CS

protocol. Suppose the process A wants to send and receive the data from process B. The following steps occur:

❖ TCP Segment Format



Where,

- **Source port address:** It is used to define the address of the application program in a source computer. It is a 16-bit field.
- **Destination port address:** It is used to define the address of the application program in a destination computer. It is a 16-bit field.
- **Sequence number:** A stream of data is divided into two or more TCP segments. The 32-bit sequence number field represents the position of the data in an original data stream.
- **Acknowledgement number:** A 32-field acknowledgement number acknowledge the data from other communicating devices. If ACK field is set to 1, then it specifies the sequence number that the receiver is expecting to receive.
- **Header Length (HLEN):** It specifies the size of the TCP header in 32-bit words. The minimum size of the header is 5 words, and the maximum size of the header is 15 words. Therefore, the maximum size of the TCP header is 60 bytes, and the minimum size of the TCP header is 20 bytes.
- **Reserved:** It is a six-bit field which is reserved for future use.
- **Control bits:** Each bit of a control field functions individually and independently. A control

bit defines the use of a segment or serves as a validity check for other fields.

There are total six types of flags in control field:

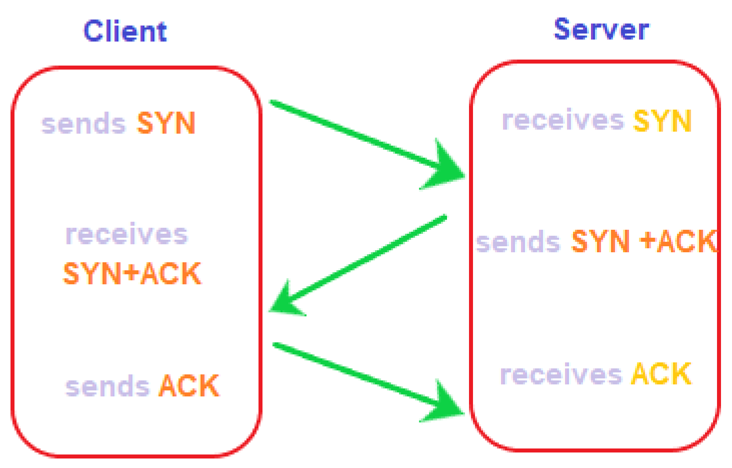
- **URG:** The URG field indicates that the data in a segment is urgent.
- **ACK:** When ACK field is set, then it validates the acknowledgement number.
- **PSH:** The PSH field is used to inform the sender that higher throughput is needed so if possible, data must be pushed with higher throughput.
- **RST:** The reset bit is used to reset the TCP connection when there is any confusion occurs in the sequence numbers.
- **SYN:** The SYN field is used to synchronize the sequence numbers in three types of segments: connection request, connection confirmation (with the ACK bit set), and confirmation acknowledgement.
- **FIN:** The FIN field is used to inform the receiving TCP module that the sender has finished sending data. It is used in connection termination in three types of segments: termination request, termination confirmation, and acknowledgement of termination confirmation.
    - **Window Size:** The window is a 16-bit field that defines the size of the window.
    - **Checksum:** The checksum is a 16-bit field used in error detection.
    - **Urgent pointer:** If URG flag is set to 1, then this 16-bit field is an offset from the sequence number indicating that it is a last urgent data byte.
    - **Options and padding:** It defines the optional fields that convey the additional information to the receiver.

❖ TCP Connection (A 3-way handshake)

Handshake refers to the process to establish connection between the client and server. Handshake is simply defined as the process to establish a communication link. To transmit a packet, TCP needs a three way handshake before it starts sending data. The reliable communication in TCP is termed as **PAR** (Positive Acknowledgement Re-transmission). When a sender sends the data to the receiver, it requires a positive acknowledgement from the receiver confirming the arrival of data. If the acknowledgement has not reached the sender, it needs to resend that data. The positive acknowledgement from the receiver establishes a successful connection.

Here, the server is the server and client is the receiver. The above diagram shows 3 steps for successful connection. A 3-way handshake is commonly known as SYN-SYN-ACK and requires both the client and server response to exchange the data. SYN means **synchronize Sequence Number** and ACK means **acknowledgment**. Each step is a type of handshake between the sender and the receiver.

The diagram of a successful TCP connection showing the three handshakes is shown below:



The three handshakes are discussed in the below steps:

Step 1: SYN

SYN is a segment sent by the client to the server. It acts as a **connection request** between the client and server. It informs the server that the client wants to establish a connection. Synchronizing sequence numbers also helps synchronize sequence numbers sent between any two devices, where the same SYN segment asks for the sequence number with the connection request.

Step 2: SYN-ACK

It is an SYN-ACK segment or an SYN + ACK segment sent by the server. The ACK segment informs the client that the server has received the connection request and it is ready to build the connection. The SYN segment informs the sequence number with which the server is ready to start with the segments.

V2V EDTECH LLP
DIPLOMA | DEGREE | BSCIT/CS

Step 3: ACK

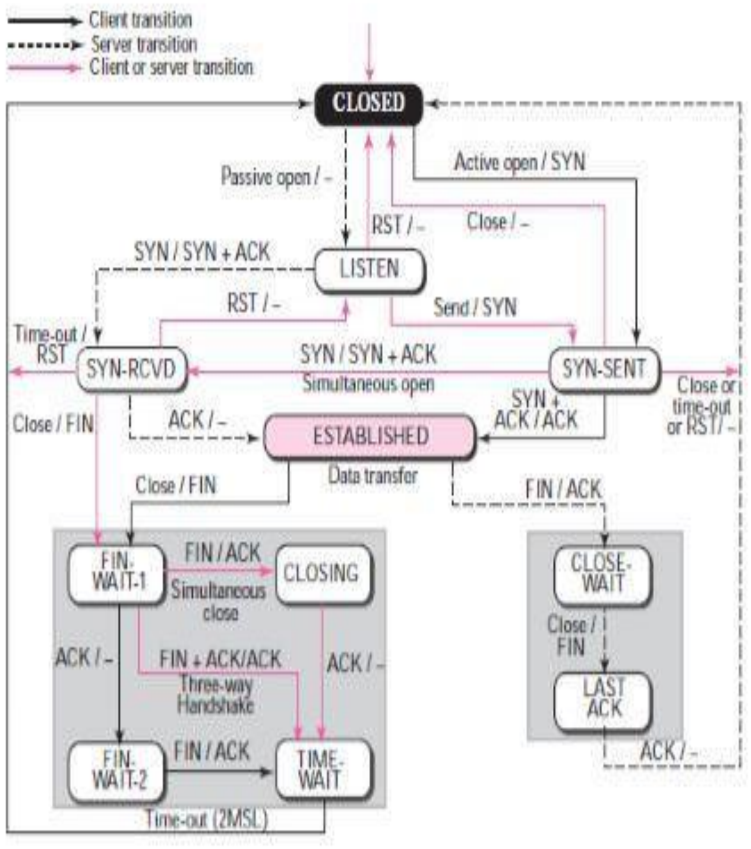
ACK (Acknowledgment) is the last step before establishing a successful TCP connection between the client and server. The ACK segment is sent by the client as the response of the received ACK and SN from the server. It results in the establishment of a reliable data connection.

After these three steps, the client and server are ready for the data communication process. TCP connection and termination are full-duplex, which means that the data can travel in both the directions simultaneously.

❖ State transaction diagram

To keep track of all the different events happening during connection establishment, connection termination, and data transfer, TCP is specified as the finite state machine shown in Figure.

Free Study Material Buy Ty Diploma Buy Sy Diploma Whatsapp Group for Study Material

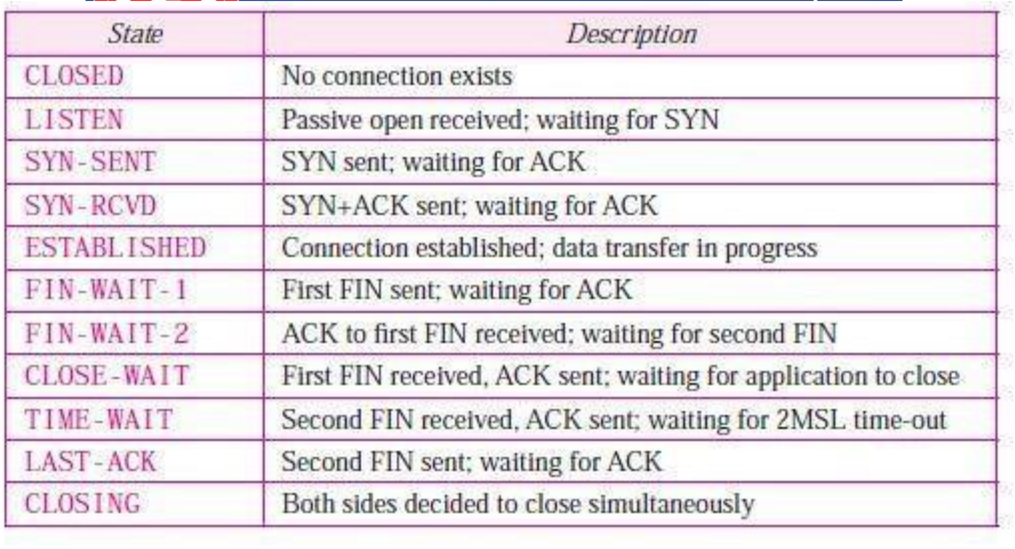


The figure shows the two FSMs used by the TCP client and server combined in one diagram. The ovals represent the states. The transition from one state to another is shown using directed lines.

The dotted black lines in the figure represent the transition that a server normally goes through; the solid black lines show the transitions that a client normally goes through.

The state marked as ESTBLISHED in the FSM is in fact two different sets of states that the client and server undergo to transfer data.

The states for TCP are as follows:



State	Description
CLOSED	No connection exists
LISTEN	Passive open received; waiting for SYN
SYN-SENT	SYN sent; waiting for ACK
SYN-RCVD	SYN+ACK sent; waiting for ACK
ESTABLISHED	Connection established; data transfer in progress
FIN-WAIT-1	First FIN sent; waiting for ACK
FIN-WAIT-2	ACK to first FIN received; waiting for second FIN
CLOSE-WAIT	First FIN received, ACK sent; waiting for application to close
TIME-WAIT	Second FIN received, ACK sent; waiting for 2MSL time-out
LAST-ACK	Second FIN sent; waiting for ACK
CLOSING	Both sides decided to close simultaneously

❖ SCTP

The Stream Control Transmission Protocol (SCTP) is a new transport protocol at the same layer as TCP and UDP. It provides functions for association management, sequence delivery, message chunk building, packet validation, and path management. SCTP is a new reliable, message-oriented transport layer protocol. It is mostly designed for Internet applications that have recently been introduced.

These new applications such as IUA (ISDN over IP), M2UA and M3UA (Telephony Signalling), H.248 (Media Gateway Control), H.323 (IP telephony), and SIP (IP Telephony), need a more sophisticated service than TCP can provide.

❖ SCTP services

The services provided by the SCTP are as follows −

- **Process-to-Process Communication** − SCTP uses all ports in the TCP space.
- **Multiple Streams** − SCTP allows multi stream service in every connection, which is called association in SCTP terminology. If any one of the streams is blocked, then the other streams can deliver their data.
- **Multihoming** − The sending and receiving host can define multiple IP addresses in each end for an association. In this approach when one path fails, another interface is ready to deliver without interruption. This fault-tolerant is used when we are sending and receiving real-time

payload like Internet telephony.

- **Full-duplex Communication** − Data can flow in both directions at the same time.

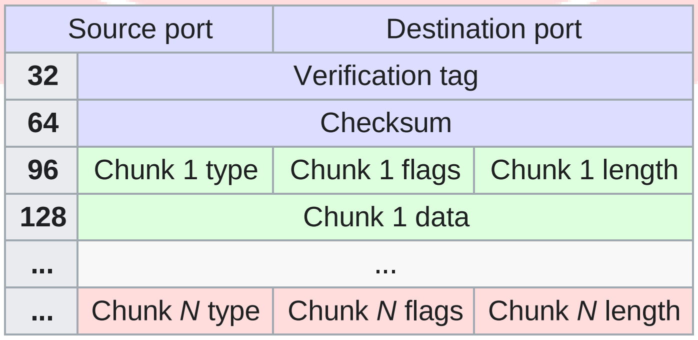
❖ Features of SCTP

The features of SCTP are as follows −

- Multihoming support.
- It is suitable for Ethernet jumbo frames because of improved error detection.
- It provides validation and acknowledgement mechanisms which protect against flooding attacks.
- It provides notification of duplicated or missing data chunks.
- It eliminates unnecessary head-of-line blocking by delivering chunks within independent data.
- It provides path selection and monitors it.
- It selects a primary data transmission path and tests its connectivity.

Stream Control Transmission Protocol (SCTP) is a connection-oriented network protocol for transmitting multiple streams of data simultaneously between two endpoints that have established a connection in a computer network.

❖ SCTP Packet Format



Source port		Destination port	
32	Verification tag		
64	Checksum		
96	Chunk 1 type	Chunk 1 flags	Chunk 1 length
128	Chunk 1 data		
...	...		
...	Chunk *N* type	Chunk *N* flags	Chunk *N* length

Free Study Material Buy Ty Diploma Buy Sy Diploma Whatsapp Group for Study Material

...	Chunk *N* data

The **Stream Control Transmission Protocol (SCTP)** has a simpler basic packet structure than TCP. Each consists of two basic sections:

1. The common header, which occupies the first 12 bytes. In the adjacent diagram, this header is highlighted in blue.
2. The data chunks, which form the remaining portion of the packet. In the diagram, the first chunk is highlighted in green and the last of N chunks (Chunk N) is highlighted in red. There are several types, including payload data and different control messages.

❖ Common header

All SCTP packets require the common header section (shown with a blue background).

Source port

This field identifies the sending port.

Destination port

This field identifies the receiving port that hosts use to route the packet to the appropriate endpoint/application.

Verification tag

A 32-bit random value created during initialization to distinguish stale packets from a previous connection.

Checksum

SCTP's original design catered for Adler-32; but RFC 3309 changed the protocol to use

the CRC32c algorithm.

❖ Chunks

Each SCTP packet consists, in addition to the common header, of chunks. Each chunk has a common format, but the contents can vary. The green bytes in the diagram above signify one chunk.

Chunk type

An 8-bit value predefined by the IETF to identify the contents of the chunk value field.

Chunk flags

V2V EDTECH LLP
DIPLOMA | DEGREE | BSCIT/CS

Eight flag-bits whose definition varies with the chunk type. The default value is zero.

Chunk length

A 16-bit unsigned value specifying the total length of the chunk in bytes (excludes any padding) that includes chunk type, flags, length, and value fields.

Chunk data

General-purpose data field whose definition varies with the chunk type.

❖ SCTP Association

Two SCTP endpoints (servers) have an SCTP association between them (rather than a TCP connection) and the SCTP service reliably transfers user messages between the peers. An

association has an association ID and includes multiple streams (unidirectional logical channels).

An upper-layer SCTP protocol (such as Diameter, for example) initiates an SCTP association, which starts a four-way handshake. The client (initiator) sends an SCTP packet with an INIT chunk which provides the server with a list of the IP addresses through which the client can be reached, a verification tag that must appear in every packet the client sends in this association (validating the sender), the number of outbound streams the client is requesting, the number of inbound streams it can support, and an initial transmission sequence number.

The server replies with an INIT-ACK chunk containing its own list of IP addresses, initial sequence number, verification tag (that must appear in every packet it sends for this association), the number of outbound streams the server is requesting, the number of inbound streams it can support, and a state cookie that ensures the association is valid.

The client then replies with a COOKIE-ECHO chunk and the server validates the cookie and replies with a COOKIE-ACK chunk. The COOKIE-ECHO and COOKIE-ACK messages can include user data (chunks) for more efficiency.

When you Configure SCTP Security, you can set an SCTP INIT timeout to control the maximum length of time after receiving an INIT chunk before the firewall receives the INIT-ACK chunk. If that time is exceeded, then the firewall stops the association initiation. You can also configure an SCTP COOKIE timeout to control the maximum length of time after

V2V EDTECH LLP
DIPLOMA | DEGREE | BSCIT/CS

receiving an INIT-ACK chunk with the STATE COOKIE before the firewall receives the COOKIE-ECHO chunk; if that time is exceeded, that also causes the firewall to stop the association initiation.

You can also leverage the following SCTP timeouts as needed:

SCTP timeout

—Maximum length of time that can elapse without SCTP traffic on an association before the firewall closes the association.

Discard SCTP timeout

—Maximum length of time that an SCTP association remains open after the firewall denies the session based on Security policy rules.

SCTP Shutdown timeout
Maximum length of time that the firewall waits after a SHUTDOWN chunk to receive a SHUTDOWN-ACK chunk before the firewall disregards the SHUTDOWN chunk.

An established SCTP association ends in one of three ways: when an endpoint sends a SHUTDOWN chunk to gracefully end the association with its peer and receives a SHUTDOWN-ACK; when an endpoint sends an ABORT chunk with or without cause parameters to close the association; or when an SCTP timeout occurs. When any of these events occur, the firewall brings down all SCTP sessions for that association.

Free Study Material   Buy Ty Diploma   Buy Sy Diploma Whatsapp Group for Study Material