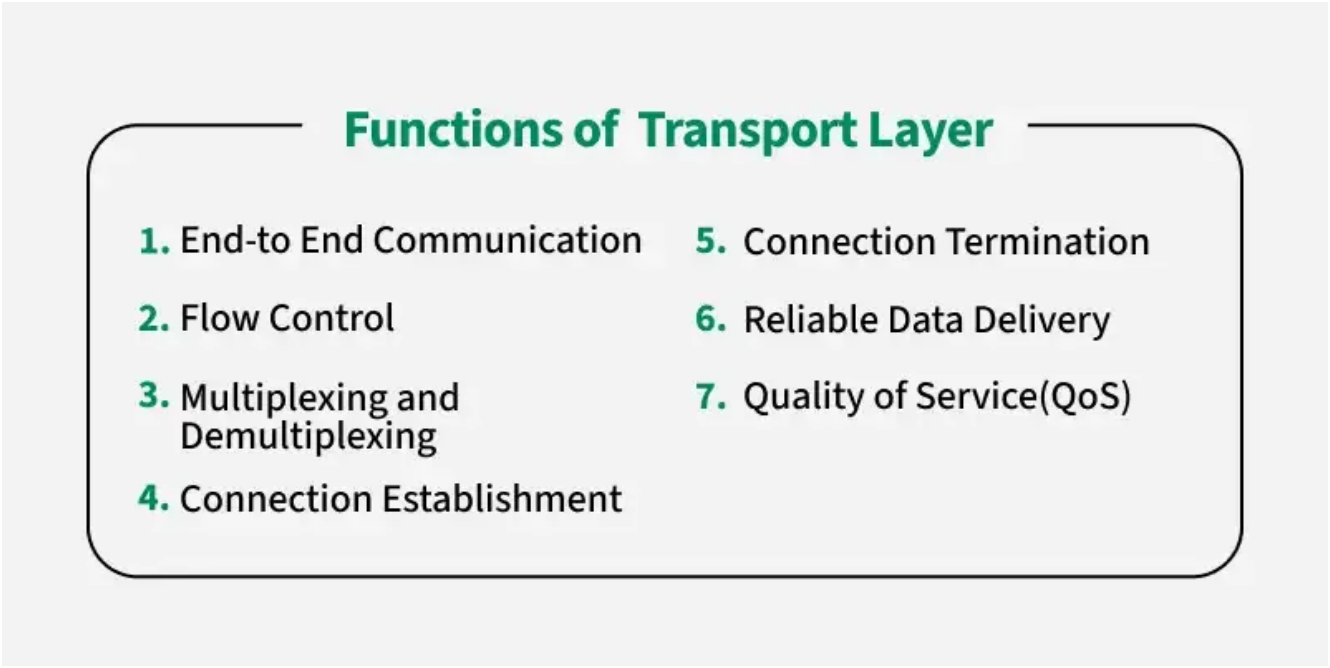
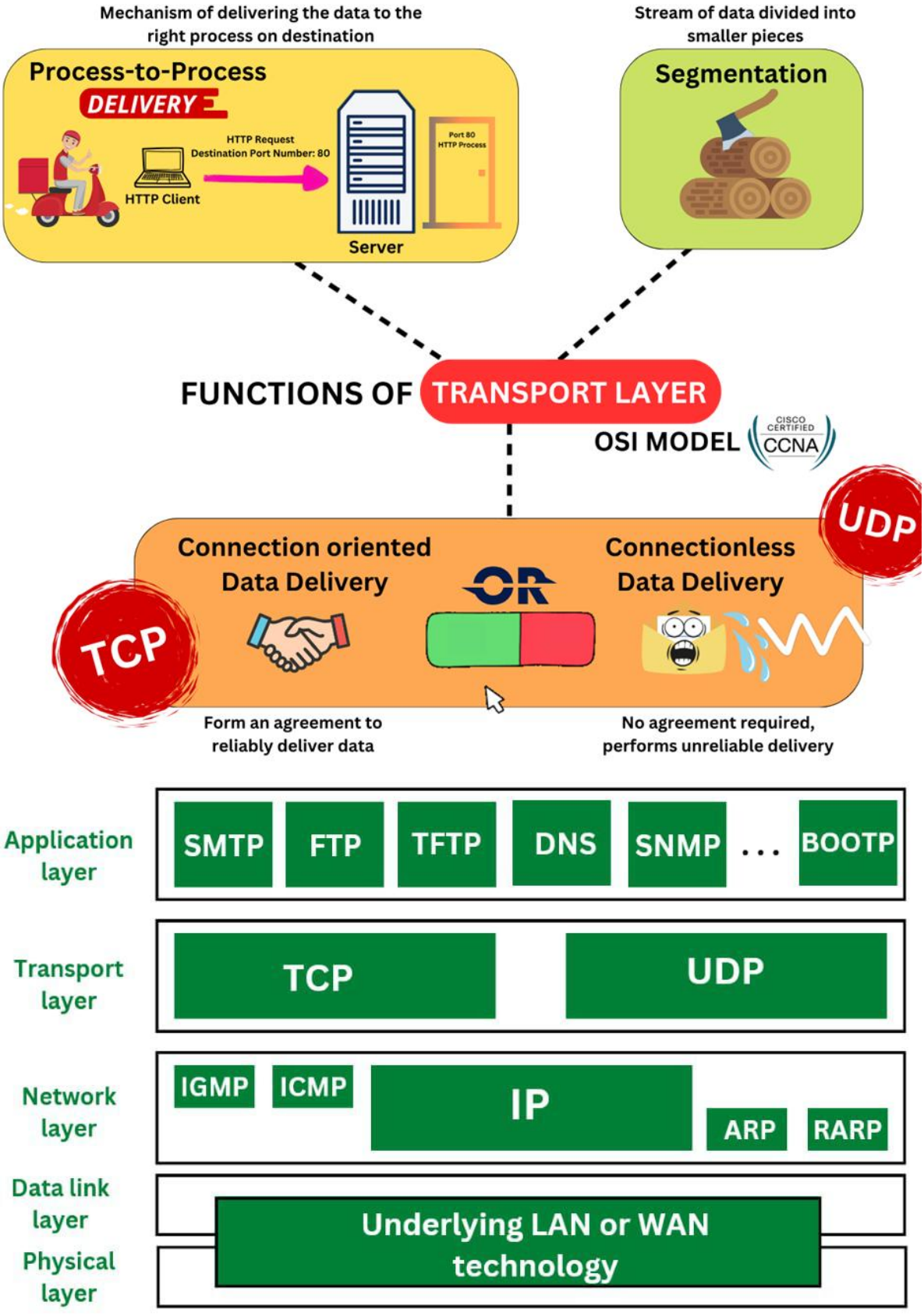
Transport layer




Mechanism of delivering the data to the right process on destination
Stream of data divided into smaller pieces
Process-to-Process
DELIVERY
HTTP Request
Destination Port Number: 80
HTTP Client
Server
Port 80
HTTP Process
Segmentation
FUNCTIONS OF
TRANSPORT LAYER
OSI MODEL
CISCO CERTIFIED CCNA
UDP
Connection oriented Data Delivery
OR
Connectionless Data Delivery
TCP
Form an agreement to reliably deliver data
No agreement required, performs unreliable delivery
Application layer
SMTP
FTP
TFTP
DNS
SNMP
...
BOOTP
Transport layer
TCP
UDP
Network layer
IGMP
ICMP
IP
ARP
RARP
Data link layer
Underlying LAN or WAN technology
Physical layer

Transport Layer Protocols

There are mainly two transport layer protocols that are used on the Internet-

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

Transmission Control Protocol-

- TCP is short for Transmission Control Protocol.
- It is a transport layer protocol.
- It has been designed to send data packets over the Internet.
- It establishes a reliable end to end connection before sending any data.

Characteristics Of TCP-

Point-01: TCP is a reliable protocol.

- It guarantees the delivery of data packets to its correct destination.
- After receiving the data packet, receiver sends an acknowledgement to the sender.
- It tells the sender whether data packet has reached its destination safely or not.
- TCP employs retransmission to compensate for packet loss.

Point-02: TCP is a connection oriented protocol.

- TCP establishes an end to end connection between the source and destination.
- The connection is established before exchanging the data.
- The connection is maintained until the application programs at each end finishes exchanging the data.

Point-03: TCP handles both congestion and flow control.

- TCP handles congestion and flow control by controlling the window size.
- TCP reacts to congestion by reducing the sender window size.

Point-04: TCP ensures in-order delivery.

- TCP ensures that the data packets get deliver to the destination in the same order they are sent by the sender.
- Sequence Numbers are used to coordinate which data has been transmitted and received.

Point-05: TCP connections are full duplex.

- TCP connection allows to send data in both the directions at the same time.
- So, TCP connections are Full Duplex.

Point-06: TCP works in collaboration with Internet Protocol.

- A TCP connection is uniquely identified by using-
- Combination of port numbers and IP Addresses of sender and receiver.
- IP Addresses indicate which systems are communicating.
- Port numbers indicate which end to end sockets are communicating.
- Port numbers are contained in the TCP header and IP Addresses are contained in the IP header.
- TCP segments are encapsulated into an IP datagram.
- So, TCP header immediately follows the IP header during transmission.

Point-07: TCP can use both selective & cumulative acknowledgements.

- TCP uses a combination of Selective Repeat and Go back N protocols.
- In TCP, sender window size = receiver window size.
- In TCP, out of order packets are accepted by the receiver.
- When receiver receives an out of order packet, it accepts that packet but sends an acknowledgement for the expected packet.
- Receiver may choose to send independent acknowledgements or cumulative acknowledgement.
- To sum up, TCP is a combination of 75% SR protocol and 25% Go back N protocol.

Point-08: TCP is a byte stream protocol.

- Application layer sends data to the transport layer without any limitation.
- TCP divides the data into chunks where each chunk is a collection of bytes.
- Then, it creates a TCP segment by adding IP header to the data chunk.
- TCP segment = TCP header + Data chunk.

Point-09: TCP provides error checking & recovery mechanism.

TCP provides error checking and recovery using three simple techniques-

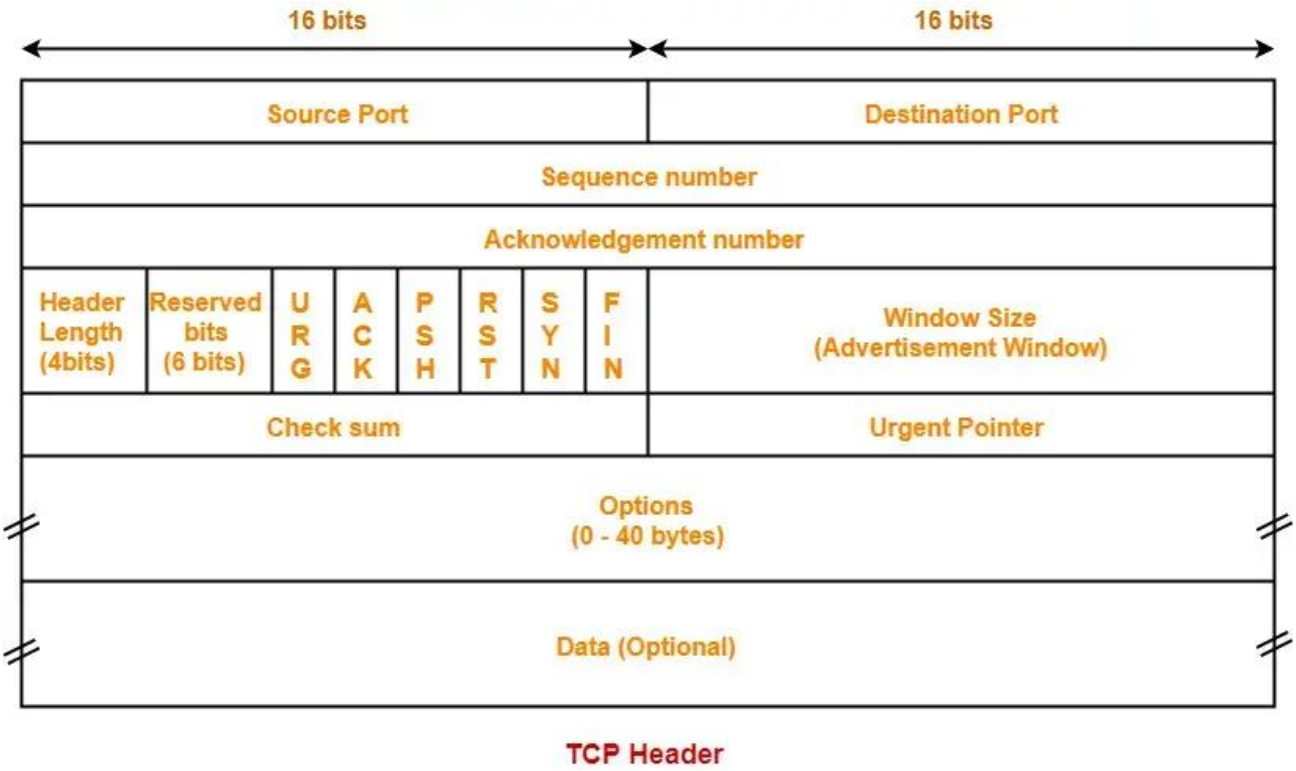
- Checksum
- Acknowledgement
- Retransmission

Note:

1. Transmission Control Protocol is a transport layer protocol.
2. It continuously receives data from the application layer.
3. It divides the data into chunks where each chunk is a collection of bytes.
4. It then creates TCP segments by adding a TCP header to the data chunks.
5. TCP segments are encapsulated in the IP datagram.

TCP segment = TCP header + Data chunk

TCP Header: the following diagram represents the TCP header format



TCP Header

1. Source Port-

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

2. Destination Port-

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

NOTE

It is important to note-

- A TCP connection is uniquely identified by using-

Combination of port numbers and IP Addresses of sender and receiver

- IP Addresses indicate which systems are communicating.
- Port numbers indicate which end to end sockets are communicating.

3. 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.

4. 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.

5. 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.

Minimum and Maximum Header length-

The length of TCP header always lies in the range-
[20 bytes , 60 bytes]

- The initial 5 rows of the TCP header are always used.
- So, minimum length of TCP header = 5 x 4 bytes = 20 bytes.

- The size of the 6th row representing the Options field vary.
- The size of Options field can go up to 40 bytes.
- So, maximum length of TCP header = 20 bytes + 40 bytes = 60 bytes.

Concept of Scaling Factor-

- Header length is a 4 bit field.
- So, the range of decimal values that can be represented is [0, 15].
- But the range of header length is [20, 60].
- So, to represent the header length, we use a scaling factor of 4.

In general,

Header length = Header length field value x 4 bytes

Examples-

- If header length field contains decimal value 5 (represented as 0101), then-

Header length = 5 x 4 = 20 bytes

- If header length field contains decimal value 10 (represented as 1010), then-

Header length = 10 x 4 = 40 bytes

- If header length field contains decimal value 15 (represented as 1111), then-

Header length = 15 x 4 = 60 bytes

6. Reserved Bits-

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

7. URG Bit-

URG bit is used to treat certain data on an urgent basis.

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.

8. ACK Bit-

ACK bit indicates whether acknowledgement number field is valid or not.

- 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**.

9. PSH Bit-

PSH bit is used to push the entire buffer immediately to the receiving application.

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.

NOTE

It is important to note-

- Unlike URG bit, PSH bit does not prioritize the data.
- It just causes all the segments in the buffer to be pushed immediately to the receiving application.
- The same order is maintained in which the segments arrived.
- It is not a good practice to set PSH bit = 1.
- This is because it disrupts the working of receiver's CPU and forces it to take an action immediately.

10. RST Bit-

RST bit is used to reset the TCP connection.

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.

11. SYN Bit-

SYN bit is used to synchronize the sequence numbers.

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.

12. FIN Bit-

FIN bit is used to terminate the TCP connection.

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.

13. 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**.

NOTE

It is important to note-

- The window size changes dynamically during data transmission.
- It usually increases during TCP transmission up to a point where congestion is detected.
- After congestion is detected, the window size is reduced to avoid having to drop packets.

14. 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.

15. 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.

16. Options-

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

Options field is generally used for the following purposes-

1. Time stamp
2. Window size extension
3. Parameter negotiation
4. Padding

A. Time Stamp-

When wrap around time is less than life time of a segment,

- Multiple segments having the same sequence number may appear at the receiver side.
- This makes it difficult for the receiver to identify the correct segment.
- If time stamp is used, it marks the age of TCP segments.
- Based on the time stamp, receiver can identify the correct segment.

B. Window Size Extension-

- Options field may be used to represent a window size greater than 16 bits.
- Using window size field of TCP header, window size of only 16 bits can be represented.

- If the receiver wants to receive more data, it can advertise its greater window size using this field.
- The extra bits are then appended in Options field.

C. Parameter Negotiation-

Options field is used for parameters negotiation.
Example- During connection establishment,

- Both sender and receiver have to specify their maximum segment size.
- To specify maximum segment size, there is no special field.
- So, they specify their maximum segment size using this field and negotiates.

D. Padding-

- Addition of dummy data to fill up unused space in the transmission unit and make it conform to the standard size is called as padding.
- Options field is used for padding.

UDP Protocol-

- UDP is short for **User Datagram Protocol**.
- It is the simplest transport layer protocol.
- It has been designed to send data packets over the Internet.
- It simply takes the datagram from the network layer, attaches its header and sends it to the user.

Characteristics of UDP-

- It is a connectionless protocol.
- It is a stateless protocol.
- It is an unreliable protocol.
- It is a fast protocol.
- It offers the minimal transport service.
- It is almost a null protocol.
- It does not guarantee in order delivery.
- It does not provide congestion control mechanism.
- It is a good protocol for data flowing in one direction.

Need of UDP-

- 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.

UDP Header-

The following diagram represents the UDP Header Format-

UDP Header

1. Source Port-

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

2. Destination Port-

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

3. Length-

- Length is a 16 bit field.
- It identifies the combined length of UDP Header and Encapsulated data.

```
Length = Length of UDP Header + Length of encapsulated data
```

4. Checksum-

- **Checksum** is a 16 bit field used for error control.

- It is calculated on UDP Header, encapsulated data and IP pseudo header.
- Checksum calculation is not mandatory in UDP.

Applications Using UDP-

Following applications use 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.

Important Notes-

Note-01:

Size of UDP Header= 8 bytes

- Unlike TCP header, the size of UDP header is fixed.
- This is because in UDP header, all the fields are of definite size.
- Size of UDP Header = Sum of the size of all the fields = 8 bytes.

Note-02:

UDP is almost a null protocol.

This is because-

- UDP provides very limited services.
- The only services it provides are checksumming of data and multiplexing by port number.

Note-03:

UDP is an unreliable protocol.

This is because-

- UDP does not guarantee the delivery of datagram to its respective user (application).
- The lost datagrams are not retransmitted by UDP.

Note-04:

Checksum calculation is not mandatory in UDP.

This is because-

- UDP is already an unreliable protocol and error checking does not make much sense.
- Also, time is saved and transmission becomes faster by avoiding to calculate it.

It may be noted-

- To disable the checksum, the field value is set to all 0's.
- If the computed checksum is zero, the field value is set to all 1's.

Note-05:

UDP does not guarantee in order delivery.

This is because-

- UDP allows out of order delivery to ensure better performance.
- If some data is lost on the way, it does not call for retransmission and keeps transmitting data.

Note-06:

Application layer can perform some tasks through UDP.

Application layer can do the following tasks through UDP-

1. Trace Route
2. Record Route
3. Time stamp

When required,

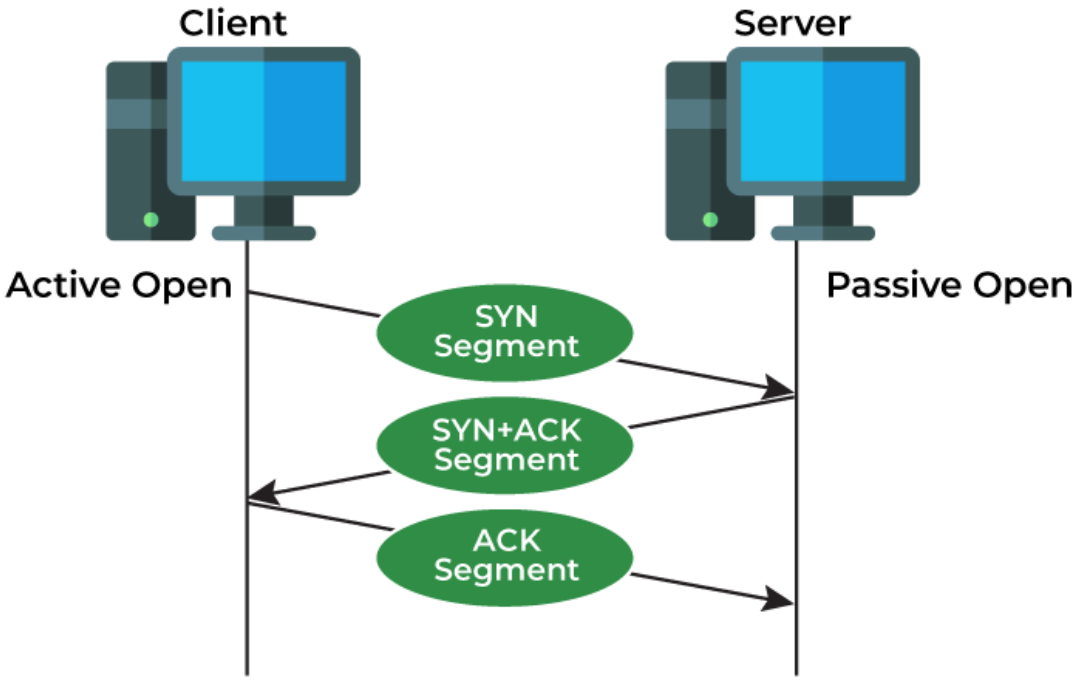
- Application layer conveys to the UDP which conveys to the IP datagram.
- UDP acts like a messenger between the application layer and the IP datagram.

Differences between TCP and UDP

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) both are protocols of the Transport Layer Protocols. TCP is a connection-oriented protocol whereas UDP is a part of the Internet Protocol suite, referred to as the UDP/IP suite. Unlike TCP, it is an unreliable and connectionless protocol. In this article, we will discuss the differences between TCP and UDP.

What is Transmission Control Protocol (TCP)?

TCP (Transmission Control Protocol) is one of the main protocols of the Internet protocol suite. It lies between the Application and Network Layers which are used in providing reliable delivery services. It is a connection-oriented protocol for communications that helps in the exchange of messages between different devices over a network. The Internet Protocol (IP), which establishes the technique for sending data packets between computers, works with TCP.



Transmission Control Protocol

Features of TCP

- TCP keeps track of the segments being transmitted or received by assigning numbers to every single one of them.

- Flow control limits the rate at which a sender transfers data. This is done to ensure reliable delivery.
- TCP implements an error control mechanism for reliable data transfer.
- TCP takes into account the level of congestion in the network.

Applications of TCP

- **World Wide Web (WWW)** : When you browse websites, TCP ensures reliable data transfer between your browser and web servers.
- **Email** : TCP is used for sending and receiving emails. Protocols like **SMTP** (Simple Mail Transfer Protocol) handle email delivery across servers.
- **File Transfer Protocol (FTP)** : FTP relies on TCP to transfer large files securely. Whether you're uploading or downloading files, TCP ensures data integrity.
- **Secure Shell (SSH)** : SSH sessions, commonly used for remote administration, rely on TCP for encrypted communication between client and server.
- **Streaming Media** : Services like Netflix, YouTube, and Spotify use TCP to stream videos and music. It ensures smooth playback by managing data segments and retransmissions.

Advantages of TCP

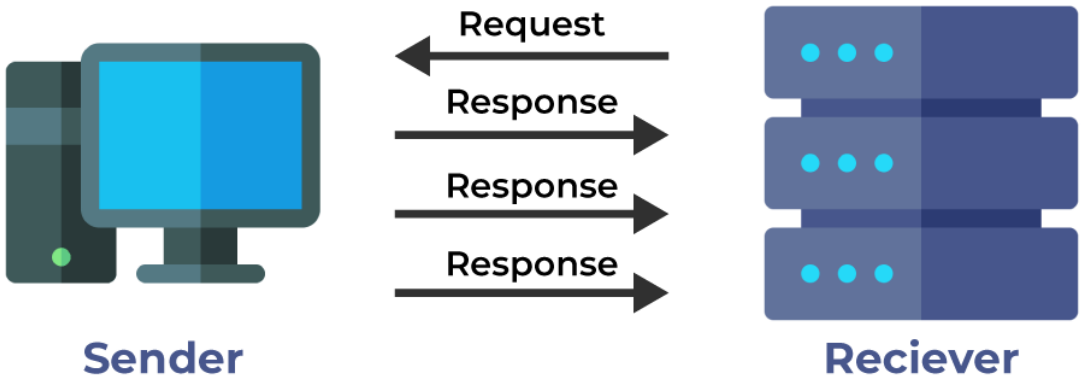
- It is reliable for maintaining a connection between Sender and Receiver.
- It is responsible for sending data in a particular sequence.
- Its operations are not dependent on Operating System .
- It allows and supports many routing protocols.
- It can reduce the speed of data based on the speed of the receiver.

Disadvantages of TCP

- It is slower than UDP and it takes more bandwidth.
- Slower upon starting of transfer of a file.
- Not suitable for LAN and PAN Networks.
- It does not have a multicast or broadcast category.
- It does not load the whole page if a single data of the page is missing.

What is User Datagram Protocol (UDP)?

User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of the Internet Protocol suite, referred to as the UDP/IP suite. Unlike TCP, it is an unreliable and connectionless protocol. So, there is no need to establish a connection before data transfer. The UDP helps to establish low-latency and loss-tolerating connections establish over the network. The UDP enables process-to-process communication.



User Datagram Protocol

Features of UDP

- Used for simple request-response communication when the size of data is less and hence there is lesser concern about flow and error control.
- It is a suitable protocol for multicasting as UDP supports packet switching .
- UDP is used for some routing update protocols like RIP(Routing Information Protocol) .
- Normally used for real-time applications which can not tolerate uneven delays between sections of a received message.

Application of UDP

- **Real-Time Multimedia Streaming** : UDP is ideal for streaming audio and video content. Its low-latency nature ensures smooth playback, even if occasional data loss occurs.
- **Online Gaming** : Many online games rely on UDP for fast communication between players.
- **DNS (Domain Name System) Queries** : When your device looks up domain names (like converting “www.example.com” to an IP address), UDP handles these requests efficiently .
- **Network Monitoring** : Tools that monitor network performance often use UDP for lightweight, rapid data exchange.
- **Multicasting** : UDP supports packet switching, making it suitable for multicasting scenarios where data needs to be sent to multiple recipients simultaneously.
- **Routing Update Protocols** : Some routing protocols, like RIP (Routing Information Protocol), utilize UDP for exchanging routing information among routers.

Advantages of UDP

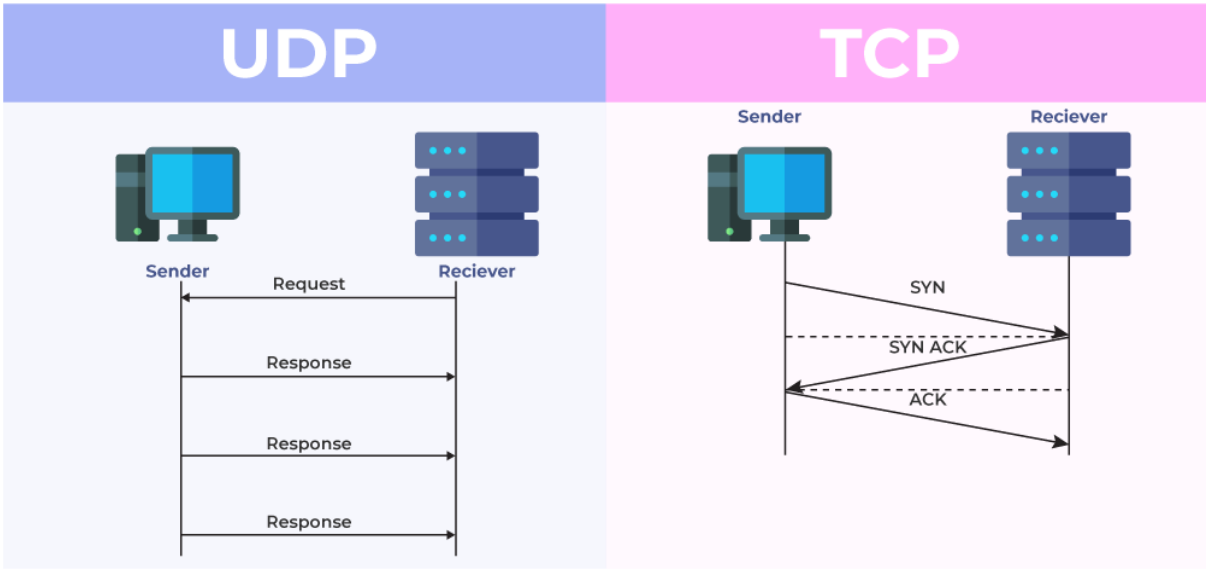
- It does not require any connection for sending or receiving data.
- Broadcast and Multicast are available in UDP.
- UDP can operate on a large range of networks.
- UDP has live and real-time data.
- UDP can deliver data if all the components of the data are not complete.

Disadvantages of UDP

- We can not have any way to acknowledge the successful transfer of data.
- UDP cannot have the mechanism to track the sequence of data.
- UDP is connectionless, and due to this, it is unreliable to transfer data.
- In case of a Collision, UDP packets are dropped by Routers in comparison to TCP.
- UDP can drop packets in case of detection of errors.

Which Protocol is Better: TCP or UDP?

The answer to this question is difficult because it totally depends on what work we are doing and what type of data is being delivered. UDP is better in the case of online gaming as it allows us to work lag-free. TCP is better if we are transferring data like photos, videos, etc. because it ensures that data must be correct has to be sent. In general, both TCP and UDP are useful in the context of the work assigned by us. Both have advantages upon the works we are performing, that's why it is difficult to say, which one is better.



Difference Between TCP and UDP

Where TCP is Used?

- Sending Emails
- Transferring Files
- Web Browsing

Where UDP is Used?

- Gaming
- Video Streaming
- Online Video Chats

Differences between TCP and UDP

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
Type of Service	TCP is a connection-oriented protocol. Connection orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data.	UDP is the Datagram-oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, or terminating a connection. UDP is efficient for broadcast and multicast types of network transmission.
Reliability	TCP is reliable as it guarantees the delivery of data to the destination router.	The delivery of data to the destination cannot be guaranteed in UDP.
Error checking mechanism	TCP provides extensive error-checking mechanisms. It is because it provides flow control and acknowledgment of data.	UDP has only the basic error-checking mechanism using checksums.
Acknowledgment	An acknowledgment segment is present.	No acknowledgment segment.
Sequence	Sequencing of data is a feature of Transmission Control Protocol (TCP). this means that packets arrive in order at the receiver.	There is no sequencing of data in UDP. If the order is required, it has to be managed by the application layer.
Speed	TCP is comparatively slower than UDP.	UDP is faster, simpler, and more efficient than TCP.
Retransmission	Retransmission of lost packets is possible in TCP, but not in UDP.	There is no retransmission of lost packets in the User Datagram Protocol (UDP).

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
Header Length	TCP has a (20-60) bytes variable length header.	UDP has an 8 bytes fixed-length header.
Weight	TCP is heavy-weight.	UDP is lightweight.
Handshaking Techniques	Uses handshakes such as SYN, ACK, SYN-ACK	It's a connectionless protocol i.e. No handshake
Broadcasting	TCP doesn't support Broadcasting.	UDP supports Broadcasting.
Protocols	TCP is used by HTTP, HTTPs , FTP , SMTP and Telnet .	UDP is used by DNS , DHCP , TFTP, SNMP , RIP , and VoIP .
Stream Type	The TCP connection is a byte stream.	UDP connection is a message stream.
Overhead	Low but higher than UDP.	Very low.
Applications	This protocol is primarily utilized in situations when a safe and trustworthy communication procedure is necessary, such as in email, on the web surfing, and in military services.	This protocol is used in situations where quick communication is necessary but where dependability is not a concern, such as VoIP, game streaming, video, and music streaming, etc.

Example: Suppose there are two houses, H1 and H2, and a letter has to be sent from H1 to H2. But there is a river in between those two houses. Now how can we send the letter?

Solution 1: Make a bridge over the river and then it can be delivered.

Solution 2: Get it delivered by a pigeon.

- Consider the first solution as **TCP** . A connection has to be made (bridge) to get the data (letter) delivered. The data is reliable because it will directly reach another end without loss of data or error.
- The second solution is **UDP** . No connection is required for sending the data. The process is fast as compared to TCP, where we need to set up a connection(bridge). But the data is not reliable: we don't know whether the pigeon will go in the right direction, will drop the letter on the way, or some issue is encountered mid-travel.

Conclusion

To summarise, TCP and UDP are both important Transport Layer protocols with distinct properties and uses. TCP offers dependable, orderly, and error-free data transmission, making it ideal for operations that require precision, such as file transfers and web browsing. UDP, on the other hand, provides quicker, connectionless communication that is excellent for real-time applications such as gaming and video streaming, when speed is critical and minor data loss is acceptable. The exact requirements of the task at hand determine whether TCP or UDP should be used.