VARDHAMAN COLLEGE OF ENGINEERING
(AUTONOMOUS)
Affiliated to **JNTUH**, Approved by **AICTE**, Accredited by **NAAC** with **A++** Grade, **ISO 9001:2015** Certified
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India.

Department of Artificial Intelligence and Machine Learning

III B.Tech I Sem (R-22)
Course: CN (A8519)

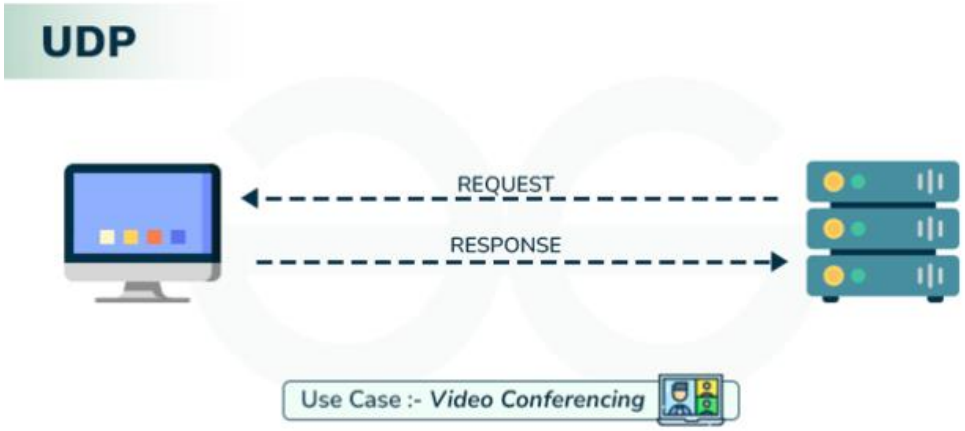
UNIT-IV

USER DATAGRAM PROTOCOL

User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of the Internet Protocol suite, referred to as 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 over the network. The UDP enables process-to-process communication.

What is User Datagram Protocol?

User Datagram Protocol (UDP) is one of the core protocols of the Internet Protocol (IP) suite. It is a communication protocol used across the internet for time-sensitive transmissions such as video playback or DNS lookups . Unlike Transmission Control Protocol (TCP), UDP is connectionless and does not guarantee delivery, order, or error checking, making it a lightweight and efficient option for certain types of data transmission.



UDP is a connectionless protocol that offers minimal error recovery services. For more insights into network protocols like UDP.

The following are the features of the 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 not wait 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.

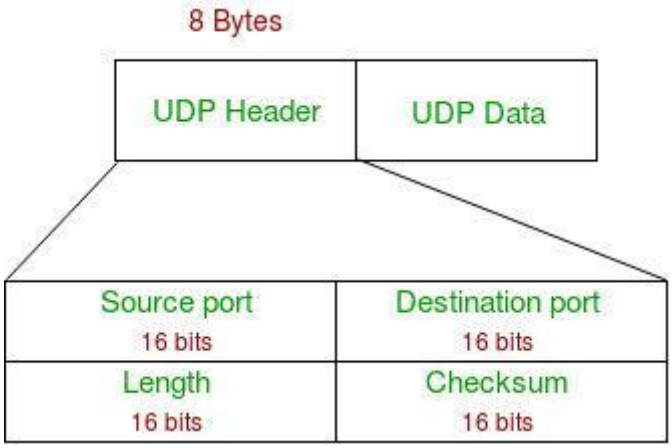
Each UDP segment is handled individually of others as each segment takes different path to reach the destination. The UDP segments can be lost or delivered out of order to reach the destination as there is no connection setup between the sender and the receiver.

Stateless

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

UDP Header

UDP header is an **8-byte** fixed and simple header, while for TCP it may vary from 20 bytes to 60 bytes. The first 8 Bytes contain all necessary header information and the remaining part consists of data. UDP port number fields are each 16 bits long, therefore the range for port numbers is defined from 0 to 65535; port number 0 is reserved. Port numbers help to distinguish different user requests or processes.



UDP Header

- **Source Port:** Source Port is a 2 Byte long field used to identify the port number of the source.
- **Destination Port:** It is a 2 Byte long field, used to identify the port of the destined packet.
- **Length:** Length is the length of UDP including the header and the data. It is a 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, the 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 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.
- VoIP (Voice over Internet Protocol) services, such as Skype and WhatsApp, use UDP for real-time voice communication. The delay in voice communication can be

noticeable if packets are delayed due to congestion control, so UDP is used to ensure fast and efficient data transmission.

- DNS (Domain Name System) also uses UDP for its query/response messages. DNS queries are typically small and require a quick response time, making UDP a suitable protocol for this application.
- DHCP (Dynamic Host Configuration Protocol) uses UDP to dynamically assign IP addresses to devices on a network. DHCP messages are typically small, and the delay caused by packet loss or retransmission is generally not critical for this application.
- Following implementations uses UDP as a transport layer protocol:
    - NTP (Network Time Protocol)
    - DNS (Domain Name Service)
    - BOOTP, DHCP.
    - NNP (Network News Protocol)
    - Quote of the day protocol
    - TFTP, RTSP, RIP.
- The application layer can do some of the tasks through UDP-
    - Trace Route
    - Record Route
    - Timestamp
- UDP takes a datagram from Network Layer , attaches its header, and sends it to the user. So, it works fast.

Advantages of UDP

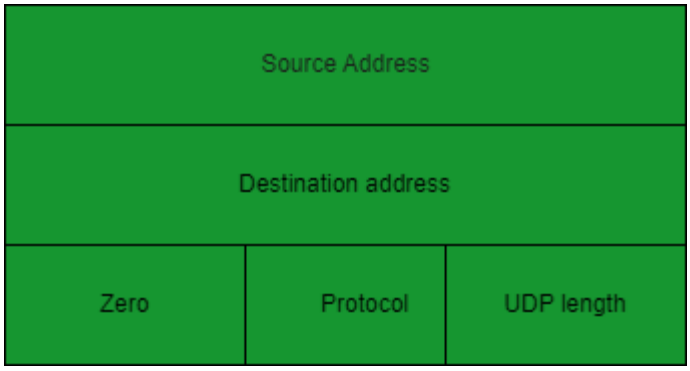
- **Speed:** UDP is faster than TCP because it does not have the overhead of establishing a connection and ensuring reliable data delivery.
- **Lower latency**: Since there is no connection establishment, there is lower latency and faster response time.
- **Simplicity:** UDP has a simpler protocol design than TCP, making it easier to implement and manage.
- **Broadcast support:** UDP supports broadcasting to multiple recipients, making it useful for applications such as video streaming and online gaming.
- **Smaller packet size:** UDP uses smaller packet sizes than TCP, which can reduce network congestion and improve overall network performance.
- User Datagram Protocol (UDP) is more efficient in terms of both latency and bandwidth.

Disadvantages of UDP

- **No reliability:** UDP does not guarantee delivery of packets or order of delivery, which can lead to missing or duplicate data.
- **No congestion control:** UDP does not have congestion control, which means that it can send packets at a rate that can cause network congestion.
- **Vulnerable to attacks:** UDP is vulnerable to denial-of-service attacks , where an attacker can flood a network with UDP packets, overwhelming the network and causing it to crash.
- **Limited use cases:** UDP is not suitable for applications that require reliable data delivery, such as email or file transfers, and is better suited for applications that can tolerate some data loss, such as video streaming or online gaming.

UDP Pseudo Header

- The purpose of using a pseudo-header is to verify that the UDP packet has reached its correct destination
- The correct destination consists of a specific machine and a specific protocol port number within that machine



UDP Pseudo Header Details

- The UDP header itself specify only protocol port number. Thus, to verify the destination UDP on the sending machine computes a checksum that covers the destination IP address as well as the UDP packet.
- At the ultimate destination, UDP software verifies the checksum using the destination IP address obtained from the header of the IP packet that carried the UDP message.
- If the checksum agrees, then it must be true that the packet has reached the intended destination host as well as the correct protocol port within that host.

User Interface

A user interface should allow the creation of new receive ports, receive operations on the receive ports that returns the data octets and an indication of source port and source address, and an operation that allows a datagram to be sent, specifying the data, source and destination ports and address to be sent.

IP Interface

- The UDP module must be able to determine the source and destination internet address and the protocol field from internet header
- One possible UDP/IP interface would return the whole internet datagram including the entire internet header in response to a receive operation
- Such an interface would also allow the UDP to pass a full internet datagram complete with header to the IP to send. the IP would verify certain fields for consistency and compute the internet header checksum.
- The IP interface allows the UDP module to interact with the network layer of the protocol stack, which is responsible for routing and delivering data across the network.
- The IP interface provides a mechanism for the UDP module to communicate with other hosts on the network by providing access to the underlying IP protocol.
- The IP interface can be used by the UDP module to send and receive data packets over the network, with the help of IP routing and addressing mechanisms.

TRANSMISSION CONTROL PROTOCOL

What is Transmission Control Protocol (TCP)?

TCP stands for **Transmission Control Protocol**. It is a transport layer protocol that facilitates the transmission of packets from source to destination. It is a connection-oriented protocol that means it establishes the connection prior to the communication that occurs between the computing devices in a network. This protocol is used with an IP protocol, so together, they are referred to as a TCP/IP.

The main functionality of the TCP is to take the data from the application layer. Then it divides the data into a several packets, provides numbering to these packets, and finally transmits these packets to the destination. The TCP, on the other side, will reassemble the packets and transmits them to the application layer. As we know that TCP is a connection-oriented protocol, so the connection will remain established until the communication is not completed between the sender and the receiver.

The following are the features of a TCP protocol:

- **Transport Layer Protocol**

TCP is a transport layer protocol as it is used in transmitting the data from the sender to the receiver.

- **Reliable**

TCP is a reliable protocol as it follows the flow and error control mechanism. It also supports the acknowledgment mechanism, which checks the state and sound arrival of the data. In the acknowledgment mechanism, the receiver sends either positive or negative acknowledgment

to the sender so that the sender can get to know whether the data packet has been received or needs to resend.

- **Order of the data is maintained**

This protocol ensures that the data reaches the intended receiver in the same order in which it is sent. It orders and numbers each segment so that the TCP layer on the destination side can reassemble them based on their ordering.

- **Connection-oriented**

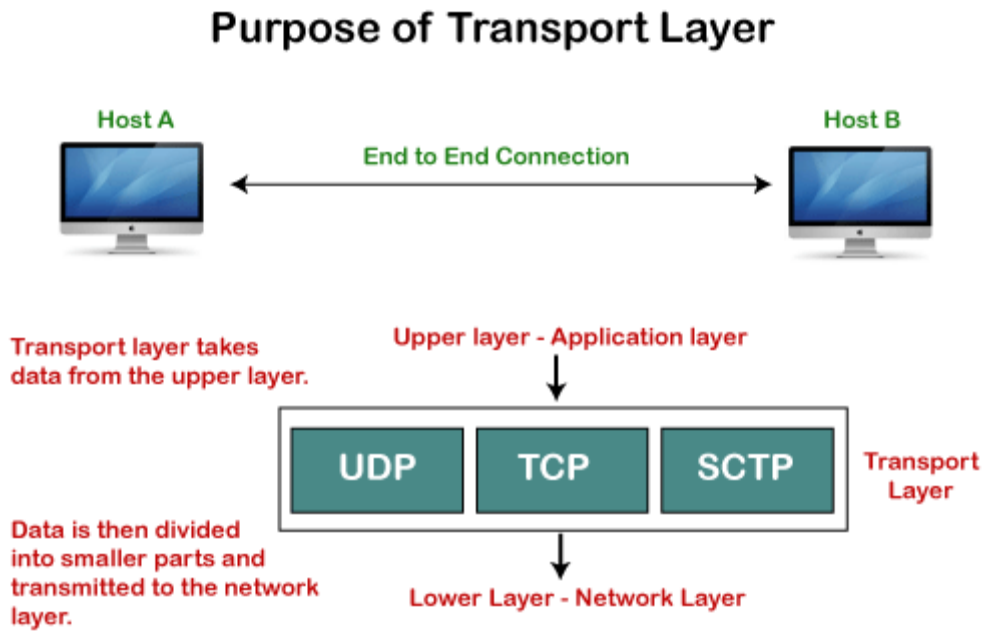
It is a connection-oriented service that means the data exchange occurs only after the connection establishment. When the data transfer is completed, then the connection will get terminated.

- **Full duplex**

It is a full-duplex means that the data can transfer in both directions at the same time.

- **Stream-oriented**

TCP is a stream-oriented protocol as it allows the sender to send the data in the form of a stream of bytes and also allows the receiver to accept the data in the form of a stream of bytes. TCP creates an environment in which both the sender and receiver are connected by an imaginary tube known as a virtual circuit. This virtual circuit carries the stream of bytes across the internet.



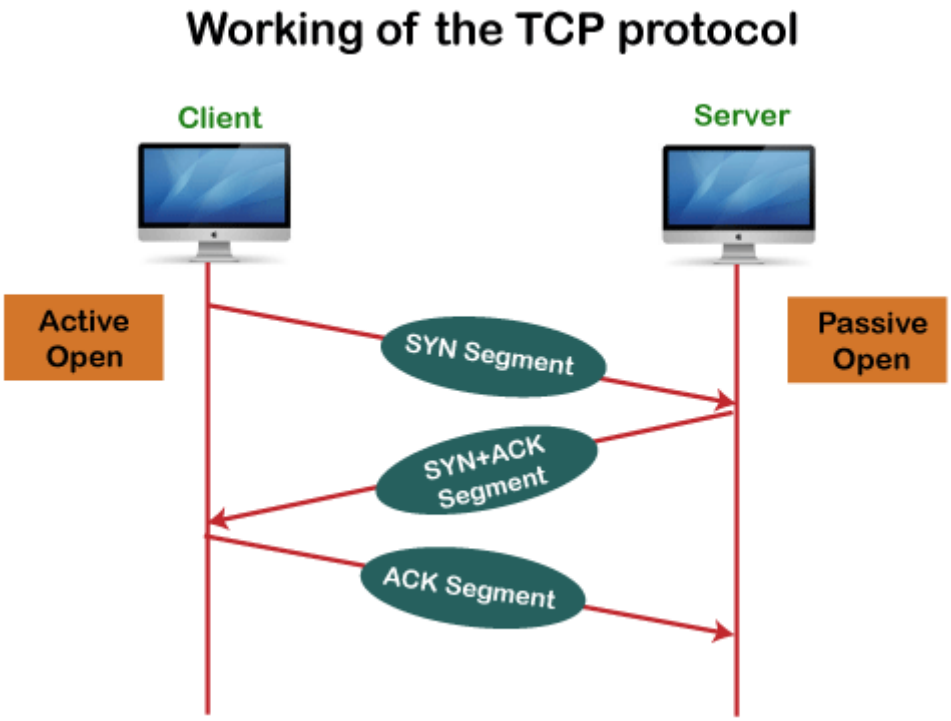
Three way handshaking mechanism in TCP

In TCP, the connection is established by using three-way handshaking.

The client sends the segment with its sequence number.

The server, in return, sends its segment with its own sequence number as well as the acknowledgement sequence, which is one more than the client sequence number.

When the client receives the acknowledgment of its segment, then it sends the acknowledgment to the server. In this way, the connection is established between the client and the server.



Synchronization Sequence Number (SYN) − The client sends the SYN to the server

- When the client wants to connect to the server, then it sends the message to the server by setting the SYN flag as 1.
- The message carries some additional information like the sequence number (32-bit random number).
- The ACK is set to 0. The maximum segment size and the window size are also set. For example, if the window size is 1000 bits and the maximum segment size is 100 bits, then a maximum of 10 data segments can be transmitted in the connection by dividing (1000/100=10).

Synchronization and Acknowledgement (SYN-ACK) to the client

- The server acknowledges the client request by setting the ACK flag to 1.
- The ACK indicates the response of the segment it received and SYN indicates with what sequence number it will start the segments.
- For example, if the client has sent the SYN with sequence number = 500, then the server will send the ACK using acknowledgment number = 5001.
- The server will set the SYN flag to '1' and send it to the client if the server also wants to establish the connection.
- The sequence number used for SYN will be different from the client's SYN.

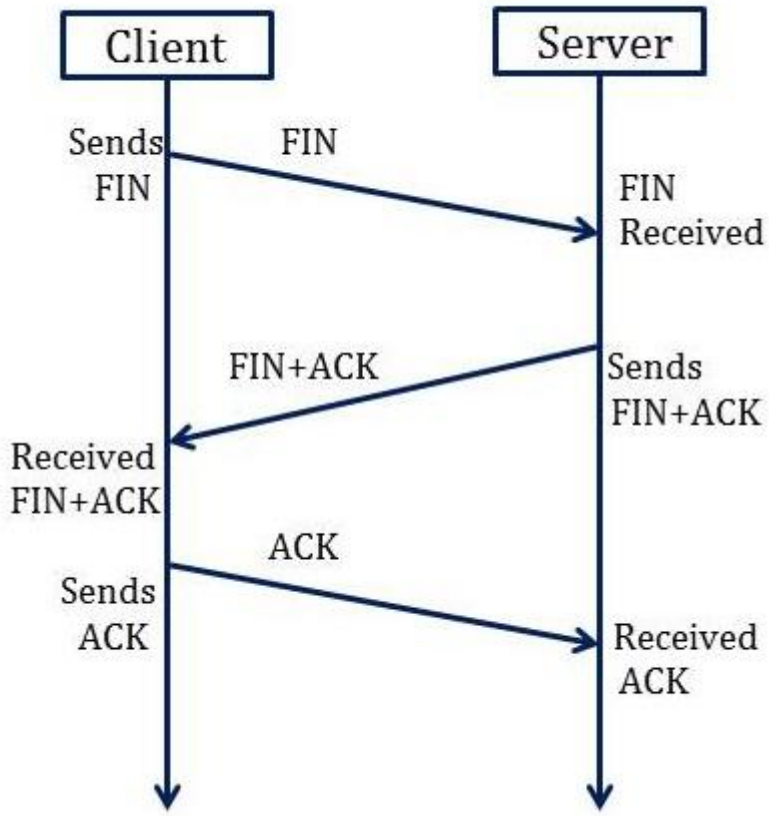
- The server also advertises its window size and maximum segment size to the client. And, the connection is established from the client-side to the server-side.

Acknowledgment (ACK) to the server

- The client sends the acknowledgment (ACK) to the server after receiving the synchronization (SYN) from the server.
- After getting the (ACK) from the client, the connection is established between the client and the server.
- Now the data can be transmitted between the client and server sides.

3 -Way Handshake Closing Connection Process to close a 3-way handshake connection

- First, the client requests the server to terminate the established connection by sending FIN.
- After receiving the client request, the server sends back the FIN and ACK request to the client.
- After receiving the FIN + ACK from the server, the client confirms by sending an ACK to the server.



Advantages of TCP

- It provides a connection-oriented reliable service, which means that it guarantees the delivery of data packets. If the data packet is lost across the network, then the TCP will resend the lost packets.
- It provides a flow control mechanism using a sliding window protocol.
- It provides error detection by using checksum and error control by using Go Back or ARP protocol.
- It eliminates the congestion by using a network congestion avoidance algorithm that includes various schemes such as additive increase/multiplicative decrease (AIMD), slow start, and congestion window.

Disadvantage of TCP

It increases a large amount of overhead as each segment gets its own TCP header, so fragmentation by the router increases the overhead.

TCP Header Format

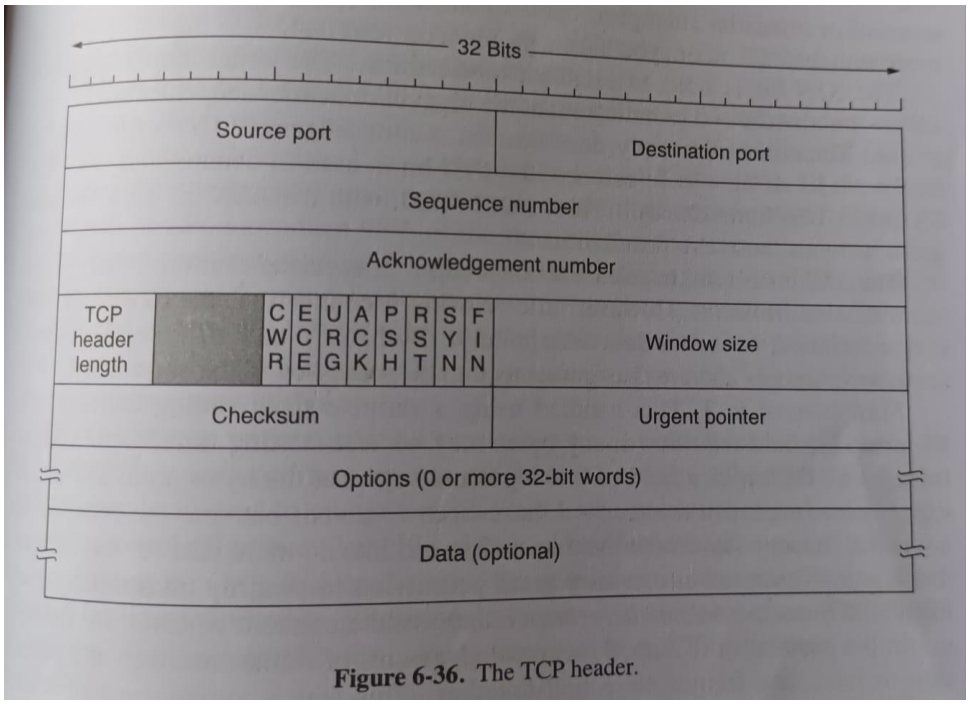


Figure 6-36. The TCP header.

- **Source port:** It defines the port of the application, which is sending the data. So, this field contains the source port address, which is 16 bits.
- **Destination port:** It defines the port of the application on the receiving side. So, this field contains the destination port address, which is 16 bits.
- **Sequence number:** This field contains the sequence number of data bytes in a particular session.

- **Acknowledgment number:** When the ACK flag is set, then this contains the next sequence number of the data byte and works as an acknowledgment for the previous data received. For example, if the receiver receives the segment number 'x', then it responds 'x+1' as an acknowledgment number.
- **HLEN:** It specifies the length of the header indicated by the 4-byte words in the header. The size of the header lies between 20 and 60 bytes. Therefore, the value of this field would lie between 5 and 15.
- **Reserved:** It is a 4-bit field reserved for future use, and by default, all are set to zero.
- **Flags**
  There are eight control bits or flags:
  - **CWR and ECE** are used to signal congestion.
  - **URG:** It represents an urgent pointer. If it is set, then the data is processed urgently.
  - **ACK:** If the ACK is set to 0, then it means that the data packet does not contain an acknowledgment.
  - **PSH:** If this field is set, then it requests the receiving device to push the data to the receiving application without buffering it.
  - **RST:** If it is set, then it requests to restart a connection.
  - **SYN:** It is used to establish a connection between the hosts.
  - **FIN:** It is used to release a connection, and no further data exchange will happen.
- **Windowsize**
  It is a 16-bit field. It contains the size of data that the receiver can accept. This field is used for the flow control between the sender and receiver and also determines the amount of buffer allocated by the receiver for a segment. The value of this field is determined by the receiver.
- **Checksum**
  It is a 16-bit field. This field is optional in UDP, but in the case of TCP/IP, this field is mandatory.
- **Urgentpointer**
  It is a pointer that points to the urgent data byte if the URG flag is set to 1. It defines a value that will be added to the sequence number to get the sequence number of the last urgent byte.
- **Options**
  It provides additional options. The optional field is represented in 32-bits. If this field contains the data less than 32-bit, then padding is required to obtain the remaining bits.

TCP vs 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.

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
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).
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, 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.