TCP vs UDP | their headers

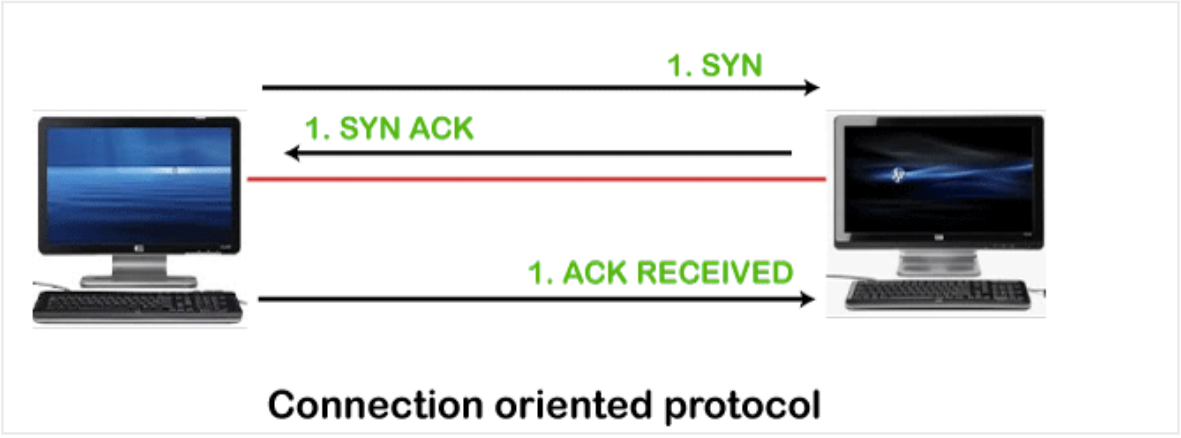
What is the TCP?

The TCP stands for **Transmission Control Protocol**. If we want the communication between two computers and communication should be good and reliable. For example, we want to view a web page, then we expect that nothing should be missing on the page, or we want to download a file, then we require a complete file, i.e., nothing should be missing either it could be a text or an image. This can only be possible due to the TCP. It is one of the most widely used protocols over the TCP/IP network.

Features of TCP

The following are the features of the TCP:

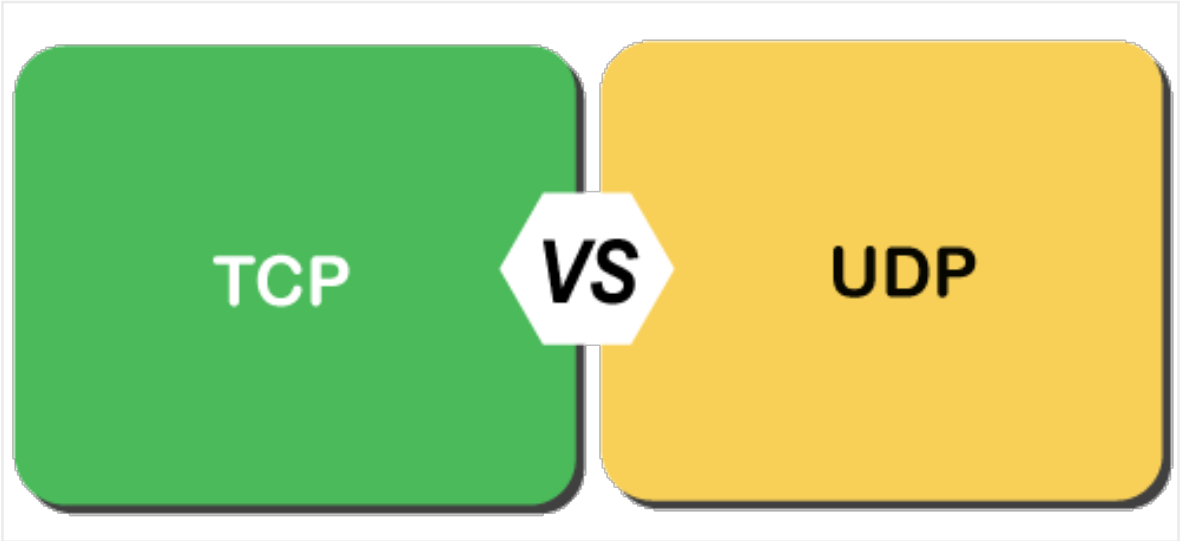
- **Data delivery**
  TCP protocol ensures that the data is received correctly, no data is missing and in order. If TCP protocol is not used, then the incorrect data can be received or out of order. For example, if we try to view the web page or download a file without using TCP, then some data or images could be missing.
- **Protocol**
  TCP is a connection-oriented protocol. Through the word **connection-oriented,** we understand that the computers first establish a connection and then do the communication. This is done by using a three-way handshake. In a **three-way handshake,** the first sender sends the SYN message to the receiver then the receiver sends back the SYN ACK message to confirm that the message has been received. After receiving the **SYN ACK** message, the sender sends the acknowledgment message to the receiver. In this way, the connection is established between the computers. Once the connection is established, the data will be delivered. This protocol guarantees the data delivery means that if the data is not received then the TCP will resend the data.



What is UDP?

The UDP stands for **User Datagram Protocol**. Its working is similar to the TCP as it is also used for sending and receiving the message. The main difference is that UDP is a connectionless protocol. Here, connectionless means that no connection establishes prior to communication. It also does not guarantee the delivery of data packets. It does not even care whether the data has been received on the receiver's end or not, so it is also known as the "fire-and-forget" protocol. It is also known as the **"fire-and-forget"** protocol as it sends the data and does not care whether the data is received or not. UDP is faster than TCP as it does not provide the assurance for the delivery of the packets.

Differences between the TCP and UDP



- **Type of protocol**
  Both the protocols, i.e., TCP and UDP, are the transport layer protocol. TCP is a connection-oriented protocol, whereas UDP is a connectionless protocol. It means that TCP requires connection prior to the communication, but the UDP does not

require any connection.

- **Reliability**
  TCP is a reliable protocol as it provides assurance for the delivery of the data. It follows the acknowledgment mechanism. In this mechanism, the sender receives the acknowledgment from the receiver and checks whether the acknowledgment is positive or negative. If the ACK is positive means, the data has been received successfully. If ACK is negative, then TCP will resend the data. It also follows the flow and error control mechanism.
  UDP is an unreliable protocol as it does not ensure the delivery of the data.
- **Flow Control**
  TCP follows the flow control mechanism that ensures a large number of packets are not sent to the receiver at the same time, while UDP does not follow the flow control mechanism.
- **Ordering**
  TCP uses ordering and sequencing techniques to ensure that the data packets are received in the same order in which they are sent. On the other hand, UDP does not follow any ordering and sequencing technique; i.e., data can be sent in any sequence.
- **Speed**
  Since TCP establishes a connection between a sender and receiver, performs error checking, and also guarantees the delivery of data packets while UDP neither creates a connection nor it guarantees the delivery of data packets, so UDP is faster than TCP.
- **Flow of data**
  In TCP, data can flow in both directions means that it provides the full-duplex service. On the other hand, UDP is mainly suitable for the unidirectional flow of data.

Let's look at the differences between the TCP and UDP in a tabular form.

	TCP	UDP
Full form	It stands for **Transmission Control Protocol**.	It stands for **User Datagram Protocol**.

Type of connection	It is a connection-oriented protocol, which means that the connection needs to be established before the data is transmitted over the network.	It is a connectionless protocol, which means that it sends the data without checking whether the system is ready to receive or not.
Reliable	TCP is a reliable protocol as it provides assurance for the delivery of data packets.	UDP is an unreliable protocol as it does not take the guarantee for the delivery of packets.
Speed	TCP is slower than UDP as it performs error checking, flow control, and provides assurance for the delivery of	UDP is faster than TCP as it does not guarantee the delivery of data packets.
Header size	The size of TCP is 20 bytes.	The size of the UDP is 8 bytes.
Acknowledgment	TCP uses the three-way-handshake concept. In this concept, if the sender receives the ACK, then the sender will send the data. TCP also has the ability to resend the lost data.	UDP does not wait for any acknowledgment; it just sends the data.
Flow control mechanism	It follows the flow control mechanism in which too many packets cannot be sent to the receiver at the same time.	This protocol follows no such mechanism.

Error checking	TCP performs error checking by using a checksum. When the data is corrected, then the data is retransmitted to the receiver.	It does not perform any error checking, and also does not resend the lost data packets.
Applications	This protocol is mainly used where a secure and reliable communication process is required, like military services, web browsing, and e-mail.	This protocol is used where fast communication is required and does not care about the reliability like VoIP, game streaming, video and music streaming, etc.

Services and Segment structure in 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 as follows:

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

2. **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 is encapsulated in an IP packet for transmission. The TCP header has information that is required for control purposes which will be discussed along with the segment structure.

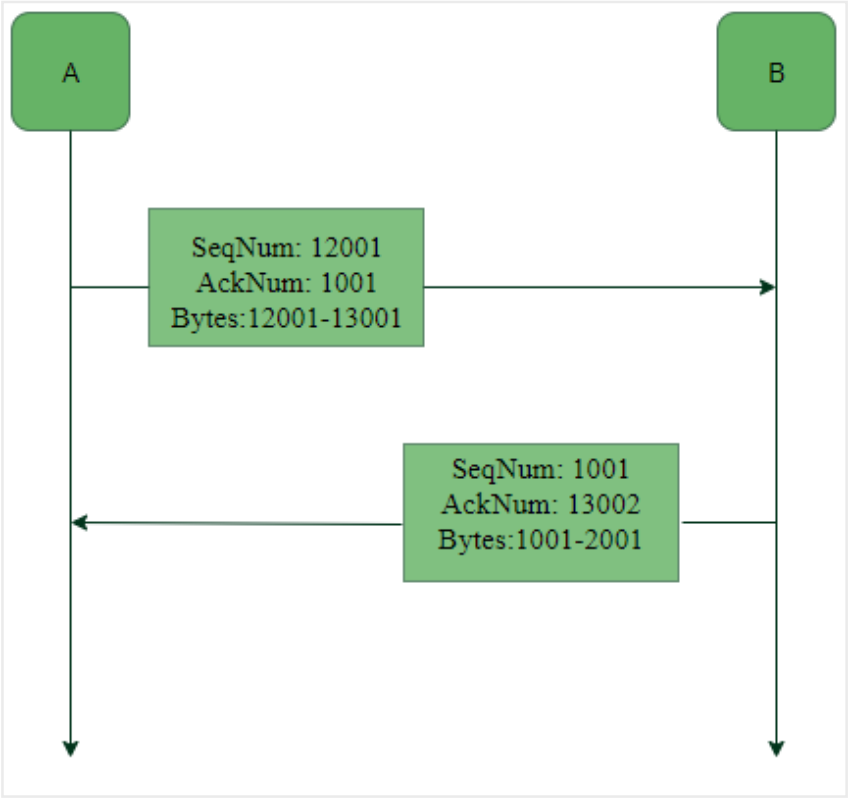
3. **Full-duplex service –**
   This means that the communication can take place in both directions at the same time.

4. **Connection-oriented service –**
   Unlike UDP, TCP provides a connection-oriented service. It defines 3 different phases:
   - Connection establishment
   - Data transfer
   - Connection termination
5. **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.

6. **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.

Byte number, Sequence number and Acknowledgement number:
All the data bytes that are to be transmitted are numbered and the beginning of this numbering is arbitrary. Sequence numbers are given to the segments so as to reassemble the bytes at the receiver end even if they arrive in a different order. The sequence number of a segment is the byte number of the first byte that is being sent. The acknowledgement number is required since TCP provides full-duplex service. The acknowledgement number is the next byte number that the receiver expects to receive which also provides

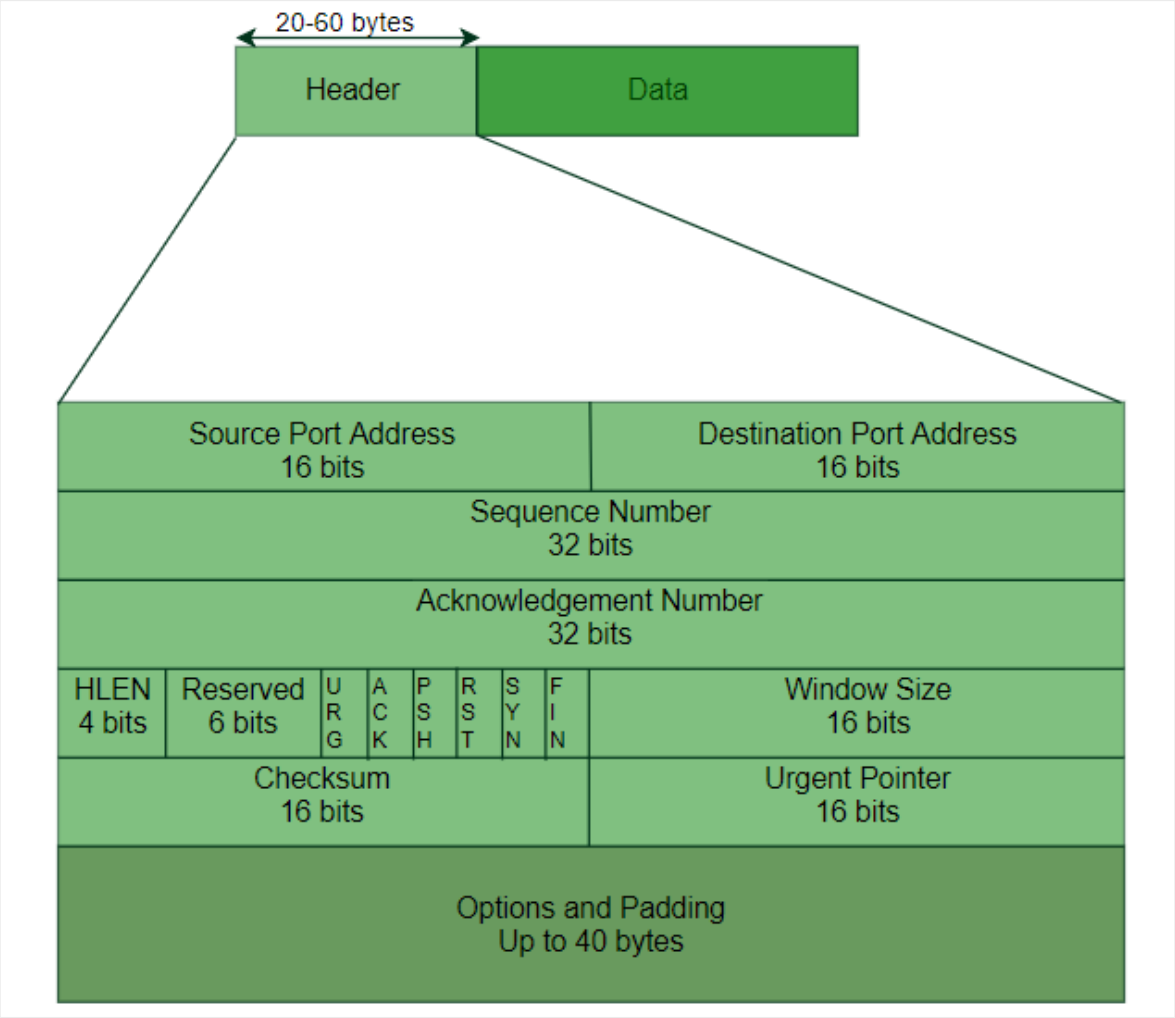
acknowledgement for receiving the previous bytes.
Example:



In this example we see that A sends acknowledgement number1001, which means that it has received data bytes till byte number 1000 and expects to receive 1001 next, hence B next sends data bytes starting from 1001. Similarly, since B has received data bytes till byte number 13001 after the first data transfer from A to B, therefore B sends acknowledgement number 13002, the byte number that it expects to receive from A next.

TCP Segment structure –

A TCP segment consists of data bytes to be sent and a header that is added to the data by TCP as shown:



The header of a TCP segment can range from 20-60 bytes. 40 bytes are for options. If there are no options, a header is 20 bytes else it can be of upmost 60 bytes.
Header fields:

- **Source Port Address –**
  A 16-bit field that holds the port address of the application that is sending the data segment.

- **Destination Port Address –**
  A 16-bit field that holds the port address of the application in the host that is receiving the data segment.

- **Sequence Number –**
  A 32-bit field that holds the sequence number, i.e, the byte number of the first byte that is sent in that particular segment. It is used to reassemble the message at the receiving end of the segments that are received out of order.

- **Acknowledgement Number –**

A 32-bit field that holds the acknowledgement number, i.e, the byte number that the receiver expects to receive next. It is an acknowledgement for the previous bytes being received successfully.

- **Header Length (HLEN) –**
  This is a 4-bit field that indicates the length of the TCP header by a number of 4-byte words in the header, i.e if the header is 20 bytes(min length of TCP header), then this field will hold 5 (because 5 × 4 = 20) and the maximum length: 60 bytes, then it'll hold the value 15(because 15 × 4 = 60). Hence, the value of this field is always between 5 and 15.

- **Control flags –**
  These are 6 1-bit control bits that control connection establishment, connection termination, connection abortion, flow control, mode of transfer etc. Their function is:
  - URG: Urgent pointer is valid
  - ACK: Acknowledgement number is valid(used in case of cumulative acknowledgement)
  - PSH: Request for push
  - RST: Reset the connection
  - SYN: Synchronize sequence numbers
  - FIN: Terminate the connection
- **Window size –**
  This field tells the window size of the sending TCP in bytes.

- **Checksum –**
  This field holds the checksum for error control. It is mandatory in TCP as opposed to UDP.

- **Urgent pointer –**
  This field (valid only if the URG control flag is set) is used to point to data that is urgently required that needs to reach the receiving process at the earliest. The value of this field is added to the sequence number to get the byte number of the last urgent byte.

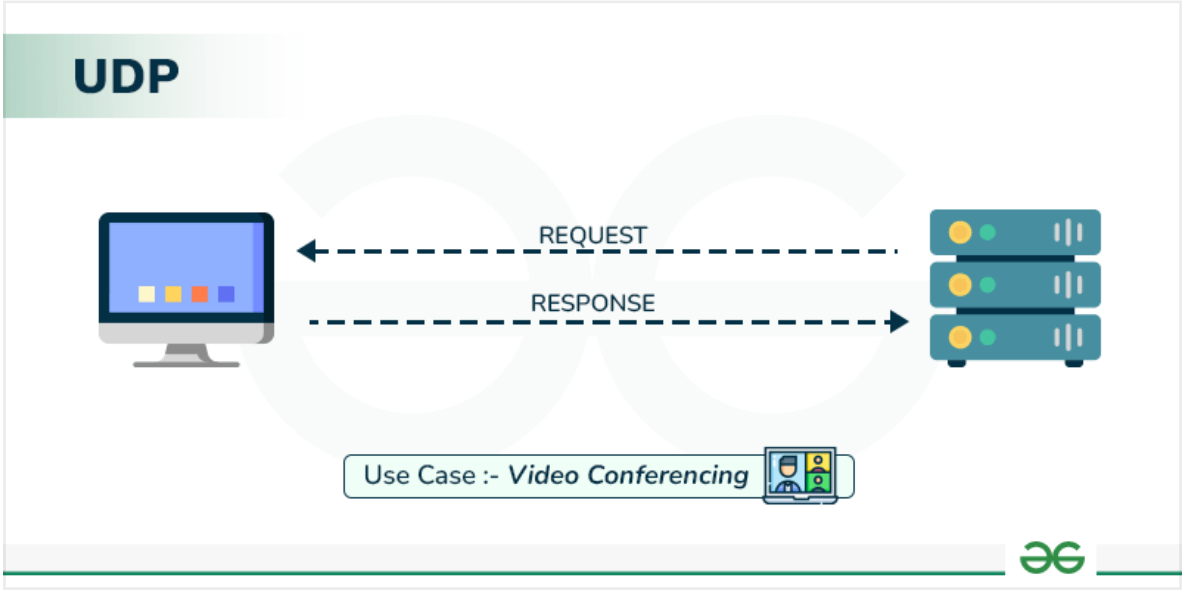
User Datagram Protocol (UDP)

Last Updated : 19 Jun, 2024

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.

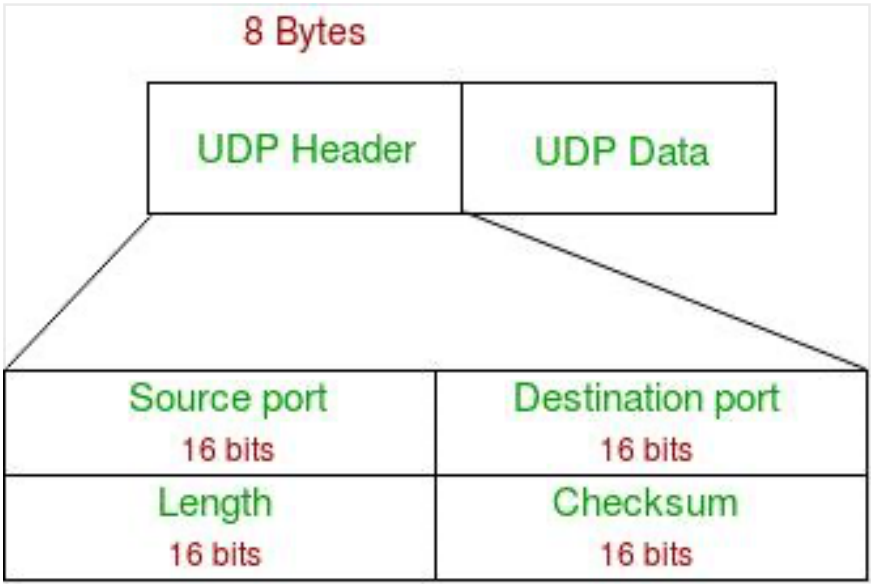
What is User Datagram Protocol?

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 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.

Notes – Unlike TCP, the Checksum calculation is not mandatory in UDP. No Error control or flow control is provided by UDP. Hence UDP depends on IP and ICMP for error reporting. Also UDP provides port numbers so that is can differentiate between users requests.

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.

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.
Acknowledgme nt	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).
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.

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.

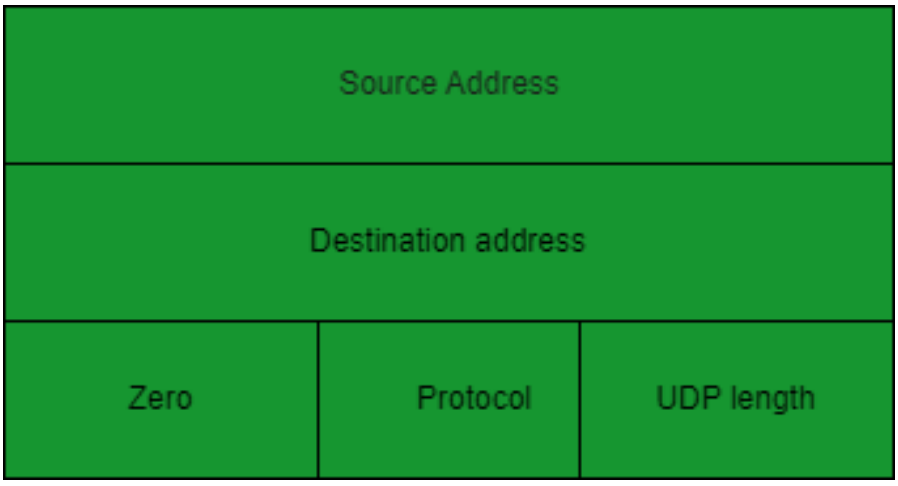
How is UDP used in DDoS attacks?

A **UDP flood attack** is a type of **Distributed Denial of Service (DDoS)** attack where an attacker sends a large number of **User Datagram Protocol (UDP)** packets to a target port.

- **UDP Protocol**: Unlike TCP, UDP is connectionless and doesn't require a handshake before data transfer. When a UDP packet arrives at a server, it checks the specified port for listening applications. If no app is found, the server sends an **ICMP** **"destination unreachable"** packet to the supposed sender (usually a random bystander due to spoofed IP addresses).
- **Attack Process**:
  - The attacker sends UDP packets with spoofed IP sender addresses to random ports on the target system.
  - The server checks each incoming packet's port for a listening application (usually not found due to random port selection).
  - The server sends ICMP "destination unreachable" packets to the spoofed sender (random bystanders).
  - The attacker floods the victim with UDP data packets, overwhelming its resources.
- **Mitigation**: To protect against UDP flood attacks, monitoring network traffic for sudden spikes and implementing security measures are crucial. Organizations often use specialized tools and services to detect and mitigate such attacks effectively.

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 consist of a specific machine and a specific protocol port number within that machine



UDP pseudo header

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.