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](https://www.javatpoint.com/ip-full-form)

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.

Features of TCP protocol

**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.

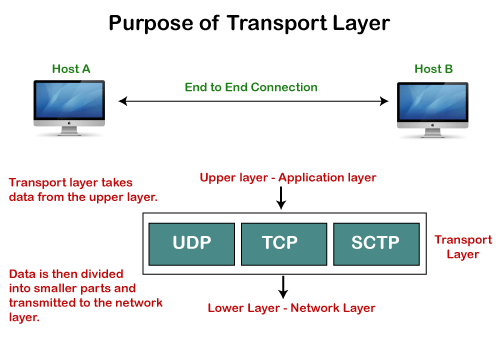
* **Stream-oriented**

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.

Need of Transport Control Protocol 0-1023

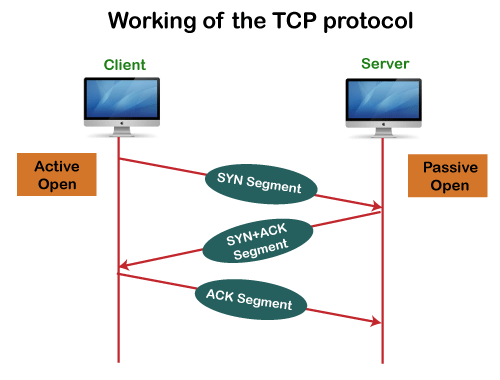
In the layered architecture of a network model, the whole task is divided into smaller tasks. Each task is assigned to a particular layer that processes the task. In the [TCP/IP model](https://www.javatpoint.com/computer-network-tcp-ip-model)

The transport layer has a critical role in providing end-to-end communication to the directly application processes. It creates 65,000 ports so that the multiple applications can be accessed at the same time. It takes the data from the upper layer, and it divides the data into smaller packets and then transmits them to the network layer.



Working of 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.



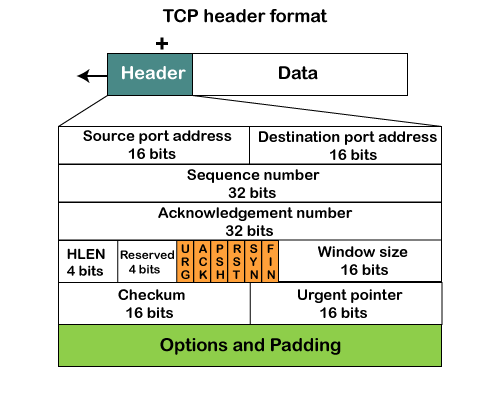
Advantages of TCP

* 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



* **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.
* **Reserved:** It is a 4-bit field reserved for future use, and by default, all are set to zero.
* **Flags**  
  **There are six control bits or flags:**
  1. **URG:** It represents an urgent pointer. If it is set, then the data is processed urgently.
  2. **ACK:** If the ACK is set to 0, then it means that the data packet does not contain an acknowledgment.
  3. **PSH:** If this field is set, then it requests the receiving device to push the data to the receiving application without buffering it.
  4. **RST:** If it is set, then it requests to restart a connection.
  5. **SYN:** It is used to establish a connection between the hosts.
  6. **FIN:** It is used to release a connection, and no further data exchange will happen.
* **Window size**  
  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.
* **Urgent pointer**  
  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.

UDP Protocol

In computer networking, the UDP stands for User Datagram Protocol. The David P. Reed developed the UDP protocol in 1980.

protocol, so it is a standard protocol over the internet. The UDP protocol allows the computer applications to send the messages in the form of datagrams from one machine to another machine over the [Internet Protocol (IP)](https://www.javatpoint.com/ip) network. The UDP is an alternative communication protocol to the TCP protocol (transmission control protocol). Like TCP, UDP provides a set of rules that governs how the data should be exchanged over the internet. The UDP works by encapsulating the data into the packet and providing its own header information to the packet. Then, this UDP packet is encapsulated to the IP packet and sent off to its destination. Both the [TCP and UDP](https://www.javatpoint.com/tcp-vs-udp)

protocols send the data over the internet protocol network, so it is also known as [TCP/IP](https://www.javatpoint.com/computer-network-tcp-ip-model)

and UDP/IP. There are many differences between these two protocols. UDP enables the process-to-process communication, whereas the TCP provides host to host communication. Since UDP sends the messages in the form of datagrams, it is considered the best-effort mode of communication.

[TCP](https://www.javatpoint.com/tcp) sends the individual packets, so it is a reliable transport medium. Another difference is that the TCP is a connection-oriented protocol whereas, the UDP is a connectionless protocol as it does not require any virtual circuit to transfer the data.

UDP also provides a different port number to distinguish different user requests and also provides the checksum capability to verify whether the complete data has arrived or not; the [IP](https://www.javatpoint.com/ip-full-form) layer does not provide these two services.

Features of UDP protocol

**The following are the features of the UDP protocol:**

* **Transport layer protocol**

[UDP](https://www.javatpoint.com/udp-full-form)

is the simplest [transport layer communication protocol](https://www.javatpoint.com/computer-network-transport-layer)

. 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.

24M

574

OOPs Concepts in Java

* **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.

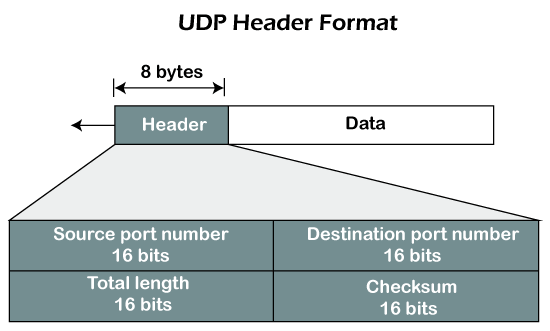
* **Stateless**

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

Why do we require the UDP protocol?

As we know that the UDP is an unreliable protocol, but we still require a UDP protocol in some cases. The UDP is deployed where the packets require a large amount of bandwidth along with the actual data. For example, in video streaming, acknowledging thousands of packets is troublesome and wastes a lot of bandwidth. In the case of video streaming, the loss of some packets couldn't create a problem, and it can also be ignored.

UDP Header Format

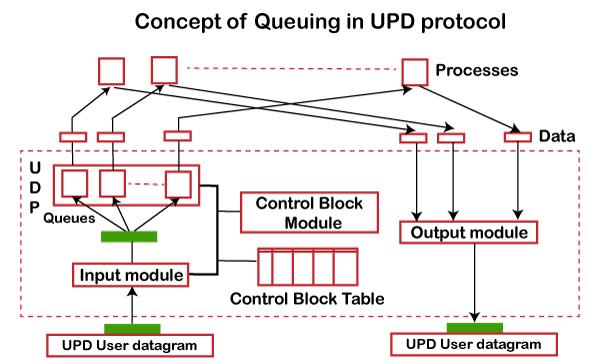


In UDP, the header size is 8 bytes, and the packet size is upto 65,535 bytes. But this packet size is not possible as the data needs to be encapsulated in the IP datagram, and an IP packet, the header size can be 20 bytes; therefore, the maximum of UDP would be 65,535 minus 20. The size of the data that 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. It is an optional field, which means that it depends upon the application, whether it wants to write the checksum or not. If it does not want to write the checksum, then all the 16 bits are zero; otherwise, it writes the checksum. In UDP, the checksum field is applied to the entire packet, i.e., header as well as data part whereas, in IP, the checksum field is applied to only the header field.

Concept of Queuing in UDP protocol



In UDP protocol, numbers are used to distinguish the different processes on a server and client. We know that UDP provides a process-to-process communication. The client generates the processes that need services while the server generates the processes that provide services. The queues are available for both the processes, i.e., two queues for each process. The first queue is the incoming queue that receives the messages, and the second one is the outgoing queue that sends the messages. The queue functions when the process is running. If the process is terminated then the queue will also get destroyed.

UDP handles the sending and receiving of the UDP packets with the help of the following components:

* **Input queue:** The UDP packets uses a set of queues for each process.
* **Input module:** This module takes the user datagram from the IP, and then it finds the information from the control block table of the same port. If it finds the entry in the control block table with the same port as the user datagram, it enqueues the data.
* **Control Block Module:** It manages the control block table.
* **Control Block Table:** The control block table contains the entry of open ports.
* **Output module:** The output module creates and sends the user datagram.

Several processes want to use the services of UDP. The UDP multiplexes and demultiplexes the processes so that the multiple processes can run on a single host.

Limitations

* It provides an unreliable connection delivery service. It does not provide any services of IP except that it provides process-to-process communication.
* The UDP message can be lost, delayed, duplicated, or can be out of order.
* It does not provide a reliable transport delivery service. It does not provide any acknowledgment or flow control mechanism. However, it does provide error control to some extent.

Advantages

* It produces a minimal number of overheads.