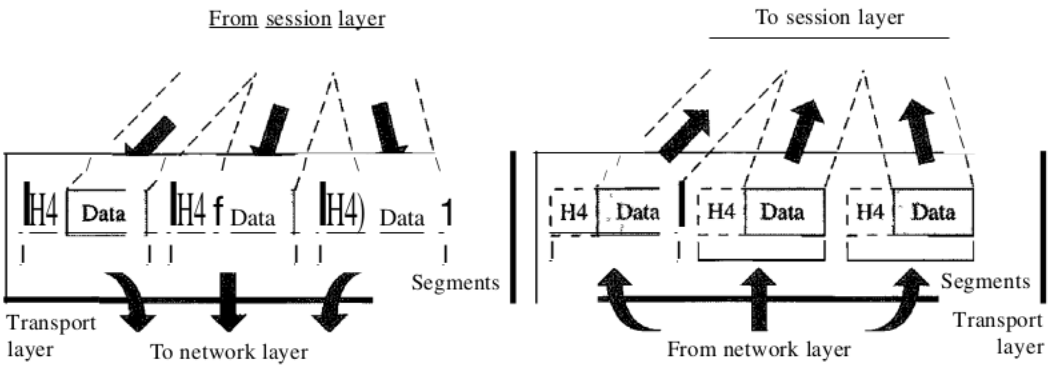
Module 7

Transport Layer

- The Transport Services
- Elements of Transport Protocols
- Congestion Control
- The Internet Transport Protocol - UDP
- The Internet Transport Protocol – TCP

Transport Layer :

- The transport layer is responsible for process-to-process delivery of the entire message.
- A process is an application program running on a host. Whereas the network layer oversees source-to-destination delivery of individual packets, it does not recognize any relationship between those packets. It treats each one independently, as though each piece belonged to a separate message, whether or not it does.
- The transport layer, on the other hand, ensures that the whole message arrives intact and in order, overseeing both error control and flow control at the source-to-destination level.



The transport layer is responsible for the delivery of a message from one process to another.

Other responsibilities of the transport layer include the following:

Service-point addressing. Computers often run several programs at the same time. For this reason, source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to a specific process (running program) on the other. The transport layer header must therefore include a type of address called a service-point address (or port address). The network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct process on that computer.

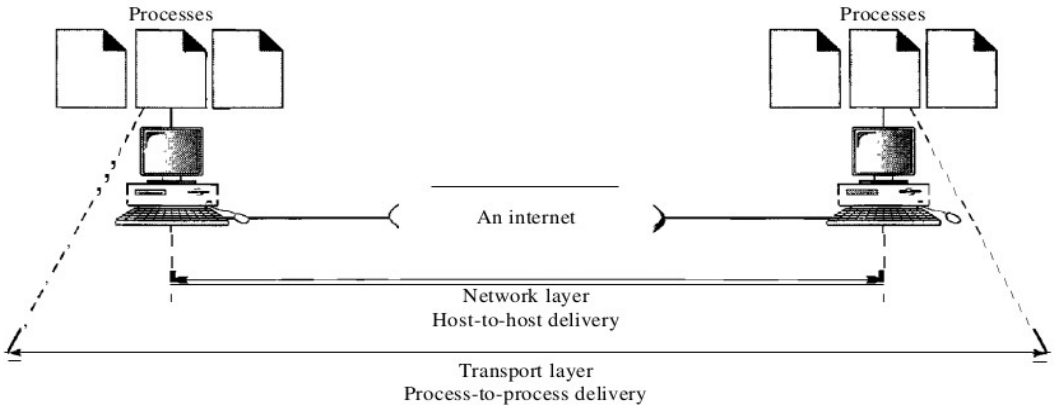
Segmentation and reassembly. A message is divided into transmittable segments, with each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in transmission.

Connection control. The transport layer can be either connectionless or connection- oriented. A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection- oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred, the connection is terminated.

Flow control. Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.

Error control. Like the data link layer, the transport layer is responsible for error control. However, error control at this layer is performed process-to- process rather than across a single link. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication). Error correction is usually achieved through retransmission.

Figure 2.11 *Reliable process-to-process delivery ofa message*



The Transport Services

• The main role of the transport layer is to provide the communication services directly to the application processes running on different hosts.

• The transport layer provides a logical communication between application processes running on different hosts. Although the application processes on different hosts are not physically connected, application processes use the logical communication provided by the transport layer to send the messages to each other.

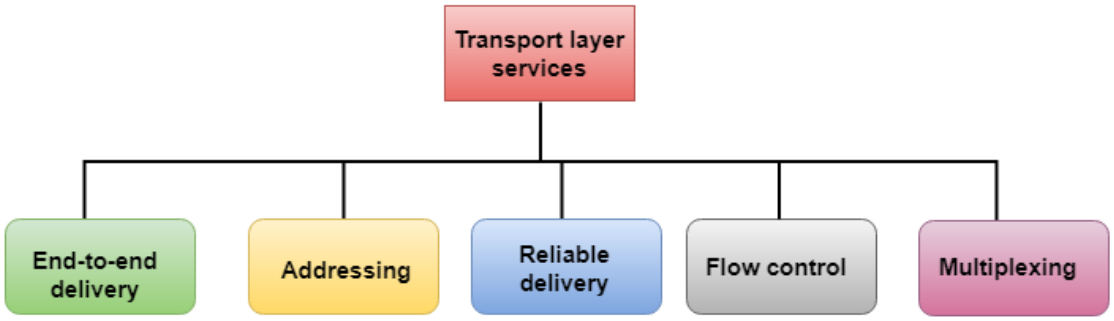
• The transport layer protocols are implemented in the end systems but not in the network routers.

• A computer network provides more than one protocol to the network applications. For example, TCP and UDP are two transport layer protocols that provide a different set of services to the network layer.

• All transport layer protocols provide multiplexing/demultiplexing service. It also provides other services such as reliable data transfer, bandwidth guarantees, and delay guarantees.

• Each of the applications in the application layer has the ability to send a message by using TCP or UDP. The application communicates by using either of these two protocols. Both TCP and UDP will then communicate with the internet protocol in the internet layer. The applications can read and write to the transport layer. Therefore, we can say that communication is a two-way process.

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



End-to-end delivery

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

Reliable delivery:

The transport layer provides reliability services by retransmitting the lost and damaged packets.
The reliable delivery has four aspects:

i. Error control

ii. Sequence control

iii. Loss control

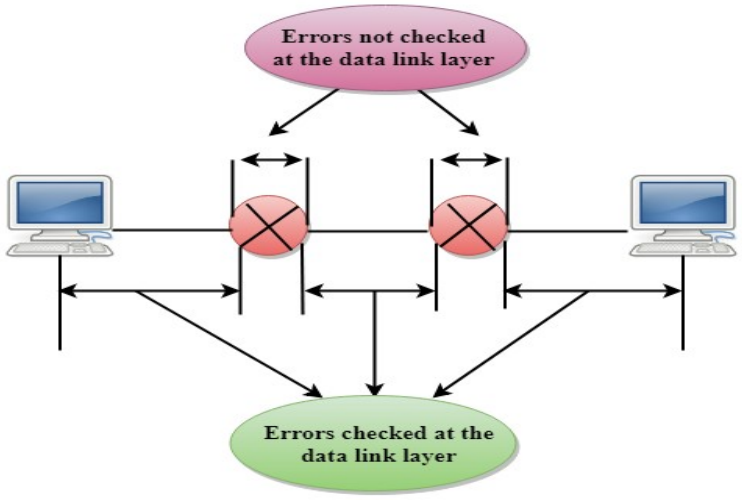
iv. Duplication control

i. Error Control

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

• The data link layer also provides the error handling mechanism, but it ensures only node-to-node error-free delivery. However, node-to-node reliability does not ensure the end-to-end reliability.

• The data link layer checks for the error between each network. If an error is introduced inside one of the routers, then this error will not be caught by the data link layer. It only detects those errors that have been introduced between the beginning and end of the link. Therefore, the transport layer performs the checking for the errors end-to-end to ensure that the packet has arrived correctly.



ii. Sequence Control

- The second aspect of the reliability is sequence control which is implemented at the transport layer.
- On the sending end, the transport layer is responsible for ensuring that the packets received from the upper layers can be used by the lower layers. On the receiving end, it ensures that the various segments of a transmission can be correctly reassembled.

iii. Loss Control

•Loss Control is a third aspect of reliability.

•The transport layer ensures that all the fragments of a transmission arrive at the destination, not some of them.

•On the sending end, all the fragments of transmission are given sequence numbers by a transport layer.

•These sequence numbers allow the receiver?s transport layer to identify the missing segment.

iv. Duplication Control

- Duplication Control is the fourth aspect of reliability.
- The transport layer guarantees that no duplicate data arrive at the destination.
- Sequence numbers are used to identify the lost packets; similarly, it allows the receiver to identify and discard duplicate segments.
- Duplication Control is the fourth aspect of reliability. The transport layer guarantees that no duplicate data arrive at the destination. Sequence numbers are used to identify the lost packets; similarly, it allows the receiver to identify and discard duplicate segments.

Flow Control

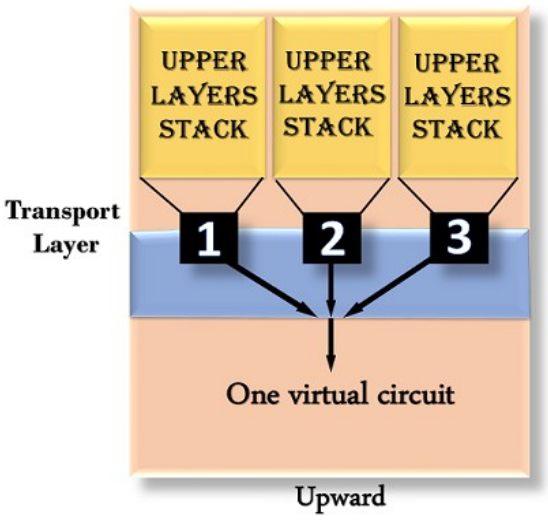
- Flow control is used to prevent the sender from overwhelming the receiver.
- If the receiver is overloaded with too much data, then the receiver discards the packets and asking for the retransmission of packets.
- This increases network congestion and thus, reducing the system performance.
- The transport layer is responsible for flow control.
- It uses the sliding window protocol that makes the data transmission more efficient as well as it controls the flow of data so that the receiver does not become overwhelmed. Sliding window protocol is byte oriented rather than frame oriented.

Multiplexing

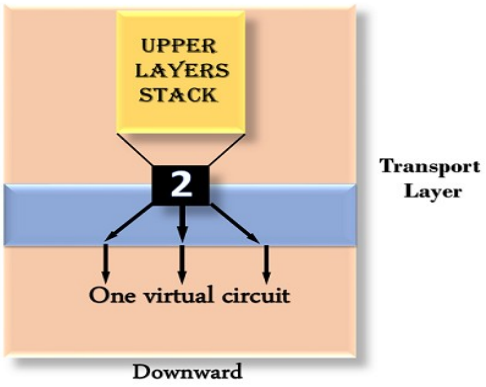
The transport layer uses the multiplexing to improve transmission efficiency.

Multiplexing can occur in two ways:

Upward multiplexing: Upward multiplexing means multiple transport layer connections use the same network connection. To make more cost-effective, the transport layer sends several transmissions bound for the same destination along the same path; this is achieved through upward multiplexing.

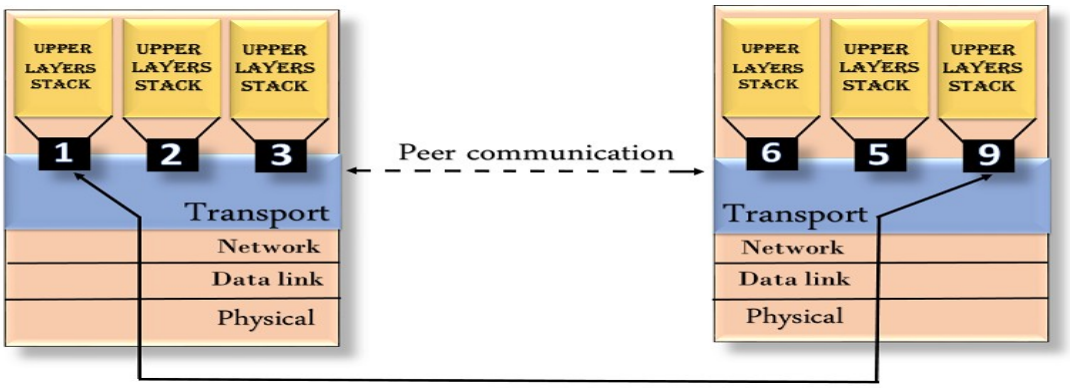


Downward multiplexing: Downward multiplexing means one transport layer connection uses the multiple network connections. Downward multiplexing allows the transport layer to split a connection among several paths to improve the throughput. This type of multiplexing is used when networks have a low or slow capacity.



Addressing

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

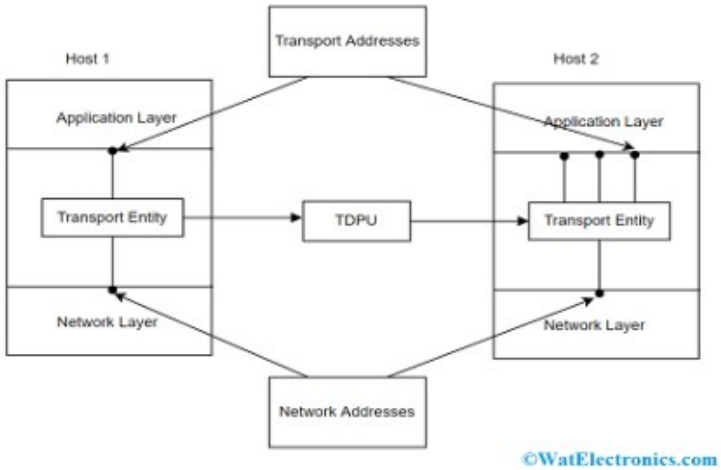


Elements of Transport Protocols

The general protocols employed by the transport layer in augmenting its functionalities are User Datagram Protocol, Datagram Congestion Control Protocol, and Transmission Control Protocol. The responsibilities/elements of the transport layer are explained as below:

i. The process to Process Delivery

- As data link layer need MAC address (containing 48-bits address in then NIC for each host device) of the source to destination hosts for proper frame delivery
- Network layer need IP address for correct packets routing, in the same approach transport later need a port number for properly delivering chunks of information to the specific process all among various processes operating on a specific host.
- The port number is a 16-bit address which is used for the identification of client and server program distinctively.



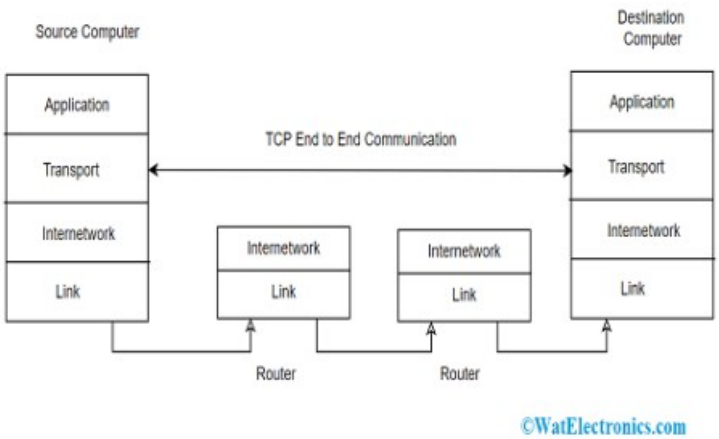
Communication Process

ii. Multiplexing & Demultiplexing

- The process of multiplexing allows synchronized usage of various applications through the network that is operating on the host.
- The transport layer offers the methodology that permits the transmission of packet streams from multiple applications at the same time on a network.
- The transport acknowledges the received packets from various processes which are differentiated by port numbers and then transmit those to the network layer after the addition of suitable headers.
- Whereas demultiplexing is the process that is performed at the receiver section in order to get the information receiving from multiple processes. Transport gets chunks of information from the network layer and then broadcasts to the specific process operating on the receiver's device.

iii. End-to-End Connection

- The transport layer also holds the responsibility of generating an end-to-end connection between various hosts and for this connection it mainly uses user datagram protocol and transmission control protocol.
- TCP is a reliable and connection-adapted protocol that makes use of handshake protocol in order to create a strong connection between the host devices.
- TCP makes sure of safe delivery of information and is implemented in multiple applications.
- On the other hand, UDP is the deported and inaccurate protocol that makes sure of good data delivery. It is applicable in situations that have little issues regarding an error or flow handling and need to transmit huge information such as video conferencing.



End to End Communication in Transport Layer

iv. Data Integrity and Error Rectification

- The information that is coming from the application layer might have errors and those are checked by the transport layer through the calculation of checksums and by detection codes.
- It analyses whether the received information is corrupted or not and through NACK and ACK services, it notifies the sender if the information has arrived or not and ensures data integrity.

v. Congestion Control

- This is the condition where multiple sources on a network try to transmit information and the router buffers start out to overflow because of packet loss.
- Due to this, packets transmitted from the sources get increases and the congestion is more increased. In this scenario, the transport layer shows the mechanism of congestion control in multiple approaches.
- It employs an open-loop type of congestion control to stop congestion and a closed-loop type of congestion control to totally eliminate the situation of congestion in a network once it takes place.

vi. Full Duplex Service

- TCP also provides the feature of Full Duplex which means that information flow takes place in both directions synchronously.
- In order to accomplish this service, every TCP needs to possess both transmission and reception buffers so that the segments flow happens in both directions.
- TCP is considered a connection-adapted protocol.

For instance, process A needs to transmit and accepts the information form process B, then the step to achieve this process are:

- Create a connection in between the TCP ends
- Information gets exchanged in both the TCP ends in both directions
- Finally, the connection is closed

vii. Flow Management

- The transport layer also offers a flow management approach in between different layers in the IP/TCP model.
- TCP even avoids data loss because of the quick sender and gradual receiver by the implementation of few flow-controlled methods.
- It even employs the sliding window protocol method that is achieved by the receiver section by transmitting a window back to the sender by notifying the information size that it can receive.

viii. Addressing

- The information which is produced by the application in one device has to be broadcasted to the suitable application on another device. To establish proper broadcasting, the transport layer provides addressing.
- The layer offers a user address that is mentioned either as a port or station. The variable of the port signifies that a specific TS customer of a corresponding station called TSAP (Transport Service access point). Every station holds only one transport entity. The protocols in this layer should be aware of the upper-layer protocols that have communication with it.

ix. Devices

- The transport layer is mainly accountable for good process-to-process communication. A few of the transport layer devices are explained below:
- Firewall – This is the device that is intended to avoid any kind of unauthorized access either to or from the private network. Few of the operations managed by firewall devices are of packet filtering and also functions as a proxy server.
- Gateways – In the domain of computer networks, a gateway is the device that is the component of two network devices that uses various protocols. It is also a protocol converter where it converts one protocol into another one. A router is considered a special type of gateway.

x. Security at Transport Layer

- The importance of Transport Layer Securities (TLS) is that they are intended to provide enhanced security at the transport layer. TLS was initially originated from a security protocol which is Secure Service Layer (SSL). It makes sure that no other external parties listen to their messages.
- The main advantages of TLS are:

Encryption – TLS helps to provide enhanced protection for the transmission of messages through encryption procedures.

Algorithm Flexibility – It offers activities for protection procedures, encryption, and hashing algorithms that are used at the time of secure sessions.

Interoperability – As TLS mostly functions with web browsers consisting of Microsoft IE and with many operating systems and internet servers.

Simple Deployment Procedures – Most of the applications such as TLS/SSL can be easily deployed on windows server 2003 OS.

Usability – As the implementation of TLS is under the application layer, many of its functionalities are totally not visible to the customers/clients.

Congestion Control

What is congestion?

A state occurring in network layer when the message traffic is so heavy that it slows down network response time.

Effects of Congestion

- As delay increases, performance decreases.
- If delay increases, retransmission occurs, making situation worse.

Congestion control algorithms

• Congestion Control is a mechanism that controls the entry of data packets into the network, enabling a better use of a shared network infrastructure and avoiding congestive collapse.

• Congestive-Avoidance Algorithms (CAA) are implemented at the TCP layer as the mechanism to avoid congestive collapse in a network.

There are two congestion control algorithm which are as follows:

Leaky Bucket Algorithm

•The leaky bucket algorithm discovers its use in the context of network traffic shaping or rate-limiting.

•A leaky bucket execution and a token bucket execution are predominantly used for traffic shaping algorithms.

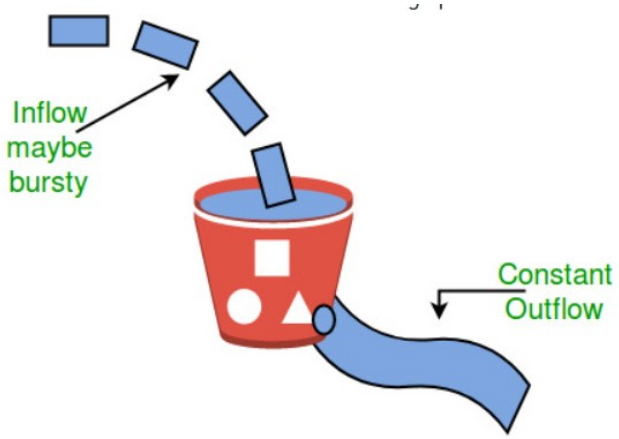
•This algorithm is used to control the rate at which traffic is sent to the network and shape the burst traffic to a steady traffic stream.

•The disadvantages compared with the leaky-bucket algorithm are the inefficient use of available network resources.

•The large area of network resources such as bandwidth is not being used effectively.

Let us consider an example to understand

Imagine a bucket with a small hole in the bottom.No matter at what rate water enters the bucket, the outflow is at constant rate.When the bucket is full with water additional water entering spills over the sides and is lost.



Similarly, each network interface contains a leaky bucket and the following steps are involved in leaky bucket algorithm:

1. When host wants to send packet, packet is thrown into the bucket.
2. The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
3. Bursty traffic is converted to a uniform traffic by the leaky bucket.
4. In practice the bucket is a finite queue that outputs at a finite rate.

Token bucket Algorithm

• The leaky bucket algorithm has a rigid output design at an average rate independent of the bursty traffic.

• In some applications, when large bursts arrive, the output is allowed to speed up. This calls for a more flexible algorithm, preferably one that never loses information. Therefore, a token bucket algorithm finds its uses in network traffic shaping or rate-limiting.

• It is a control algorithm that indicates when traffic should be sent. This order comes based on the display of tokens in the bucket.

• The bucket contains tokens. Each of the tokens defines a packet of predetermined size. Tokens in the bucket are deleted for the ability to share a packet.

• When tokens are shown, a flow to transmit traffic appears in the display of tokens.

• No token means no flow sends its packets. Hence, a flow transfers traffic up to its peak burst rate in good tokens in the bucket.

Need of token bucket Algorithm:-

The leaky bucket algorithm enforces output pattern at the average rate, no matter how bursty the traffic is. So in order to deal with the bursty traffic we need a flexible algorithm so that the data is not lost. One such algorithm is token bucket algorithm.

The Internet Transport Protocol – UDP (User Datagram Protocol)

- The David P. Reed developed the UDP protocol in 1980. It is defined in RFC 768, and it is a part of the TCP/IP 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) network.
- The UDP is an alternative communication protocol to the TCP protocol (transmission control protocol).
- 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.
- UDP enables the process to process communication.
- Since UDP sends the messages in the form of datagrams, it is considered the best-effort mode of communication.
- 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 layer does not provide these two services.

Features of UDP protocol

The following are the features of the UDP protocol:

i. 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.

ii. 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.

iii. 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.

iv. 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.

v. 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.

vi. 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.

vii. Stateless

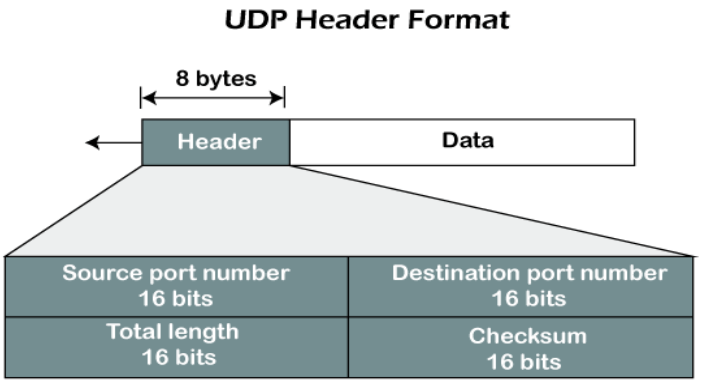
It is a stateless protocol that means that the sender does not get the acknowledgement for the packet which has been sent.

Why do we require the UDP protocol?

UDP is an unreliable protocol, but 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.



The UDP header contains four fields:

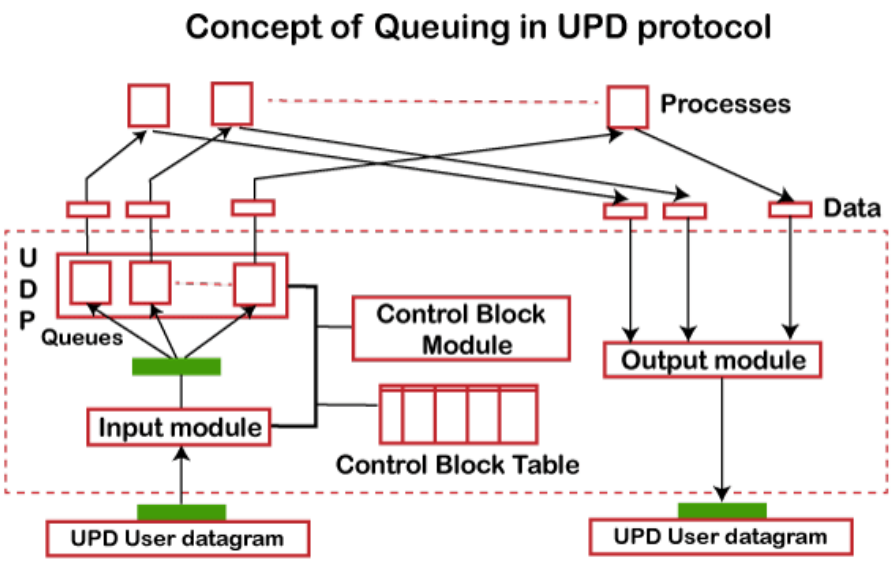
i. Source port number: It is 16-bit information that identifies which port is going t send the packet.

ii. 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.

iii. 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.

iv. 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.

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.

The Internet Transport Protocol – TCP

- TCP stands for Transmission Control Protocol.

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

Features Of TCP protocol

i. Stream data transfer: TCP protocol transfers the data in the form of contiguous stream of bytes. TCP group the bytes in the form of TCP segments and then passed it to the IP layer for transmission to the destination. TCP itself segments the data and forward to the IP.

ii. Reliability: TCP assigns a sequence number to each byte transmitted and expects a positive acknowledgement from the receiving TCP. If ACK is not received within a timeout interval, then the data is retransmitted to the destination.The receiving TCP uses the sequence number to reassemble the segments if they arrive out of order or to eliminate the duplicate segments.

Iii. Flow Control: When receiving TCP sends an acknowledgement back to the sender indicating the number the bytes it can receive without overflowing its internal buffer. The number of bytes is sent in ACK in the form of the highest sequence number that it can receive without any problem. This mechanism is also referred to as a window mechanism.

iv. Multiplexing: Multiplexing is a process of accepting the data from different applications and forwarding to the different applications on different computers. At the receiving end, the data is forwarded to the correct application. This process is known as demultiplexing. TCP transmits the packet to the correct application by using the logical channels known as ports.

v. Logical Connections: The combination of sockets, sequence numbers, and window sizes, is called a logical connection. Each connection is identified by the pair of sockets used by sending and receiving processes.

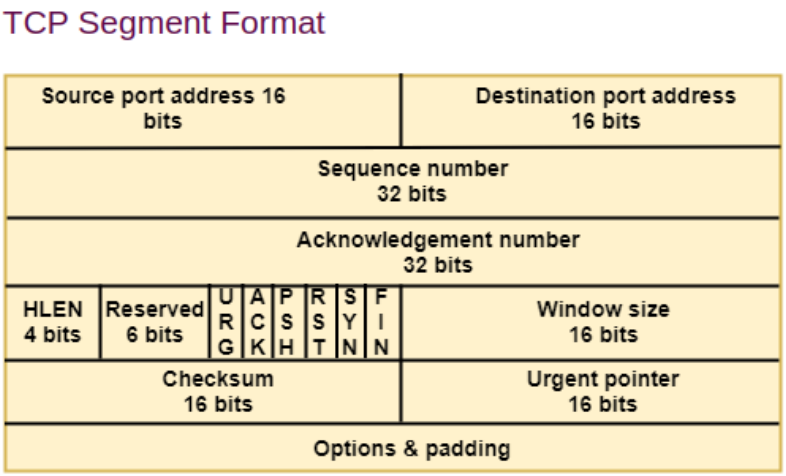
vi. Full Duplex: TCP provides Full Duplex service, i.e., the data flow in both the directions at the same time. To achieve Full Duplex service, each TCP should have sending and receiving buffers so that the segments can flow in both the directions. TCP is a connection-oriented protocol.

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

- Establish a connection between two TCPs.
- Data is exchanged in both the directions.

- The Connection is terminated.

TCP Segment Format



Where,

Source port address: It is used to define the address of the application program in a source computer. It is a 16-bit field.

Destination port address: It is used to define the address of the application program in a destination computer. It is a 16-bit field.

Sequence number: A stream of data is divided into two or more TCP segments. The 32-bit sequence number field represents the position of the data in an original data stream.

Acknowledgement number: A 32-field acknowledgement number acknowledge the data from other communicating devices. If ACK field is set to 1, then it specifies the sequence number that the receiver is expecting to receive.

Header Length (HLEN): It specifies the size of the TCP header in 32-bit words. The minimum size of the header is 5 words, and the maximum size of the header is 15 words. Therefore, the maximum size of the TCP header is 60 bytes, and the minimum size of the TCP header is 20 bytes.

Reserved: It is a six-bit field which is reserved for future use.

Control bits: Each bit of a control field functions individually and independently. A control bit defines the use of a segment or serves as a validity check for other fields.

There are total six types of flags in control field:

- URG: The URG field indicates that the data in a segment is urgent.
- ACK: When ACK field is set, then it validates the acknowledgement number.
- PSH: The PSH field is used to inform the sender that higher throughput is needed so if possible, data must be pushed with higher throughput.
- RST: The reset bit is used to reset the TCP connection when there is any confusion occurs in the sequence numbers.
- SYN: The SYN field is used to synchronize the sequence numbers in three types of segments: connection request, connection confirmation (with the ACK bit set), and confirmation acknowledgement.
- FIN: The FIN field is used to inform the receiving TCP module that the sender has finished sending data. It is used in connection termination in three types of segments: termination request, termination confirmation, and acknowledgement of termination confirmation.

Window Size: The window is a 16-bit field that defines the size of the window.

Checksum: The checksum is a 16-bit field used in error detection.

Urgent pointer: If URG flag is set to 1, then this 16-bit field is an offset from the sequence number indicating that it is a last urgent data byte.

Options and padding: It defines the optional fields that convey the additional information to the receiver.

Differences b/w TCP & UDP

Basis for Comparison	TCP	UDP
Definition	TCP establishes a virtual circuit before transmitting the data.	UDP transmits the data directly to the destination computer without verifying whether the receiver is ready to receive or not.
Connection Type	It is a Connection-Oriented	It is a Connectionless protocol

	protocol	
Speed	slow	high
Reliability	It is a reliable protocol.	It is an unreliable protocol.
Header size	20 bytes	8 bytes
acknowledgement	It waits for the acknowledgement of data and has the ability to resend the lost packets.	It neither takes the acknowledgement, nor it retransmits the damaged frame.

Reference Link

https://www.tutorialspoint.com/data_communication_computer_network transport_layer_introduction.htm

https://www.studytonight.com/computer-networks/transport-layer-in-computer-networks

https://www.tutorialspoint.com/what-is-congestion-control-algorithm

https://www.tutorialspoint.com/data_communication_computer_network/ user_datagram_protocol.htm

https://www.javatpoint.com/udp-protocol