Module 3

Network layer services

The services which are offered by the network layer protocol are as follows:

1. Packetizing
2. Routing
3. Forwarding

1. Packetizing

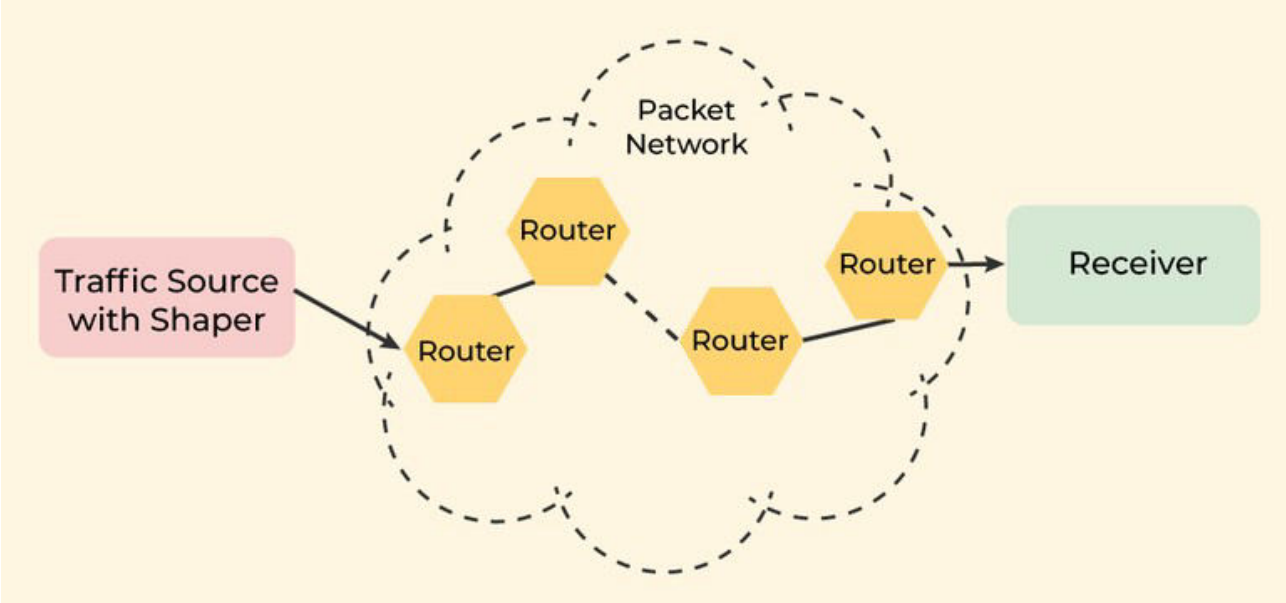
The process of encapsulating the data received from the upper layers of the network (also called payload) in a network layer packet at the source and decapsulating the payload from the network layer packet at the destination is known as packetizing.

The source host adds a header that contains the source and destination address and some other relevant information required by the network layer protocol to the payload received from the upper layer protocol and delivers the packet to the data link layer.

The destination host receives the network layer packet from its data link layer, decapsulates the packet, and delivers the payload to the corresponding upper layer protocol. The routers in the path are not allowed to change either the source or the destination address. The routers in the path are not allowed to decapsulate the packets they receive unless they need to be fragmented.

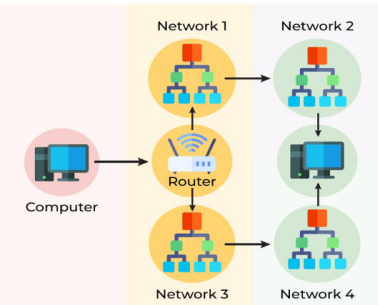
The source host adds a header that contains the source and destination address and some other relevant information required by the network layer protocol to the payload received from the upper layer protocol and delivers the packet to the data link layer.

The destination host receives the network layer packet from its data link layer, decapsulates the packet, and delivers the payload to the corresponding upper layer protocol. The routers in the path are not allowed to change either the source or the destination address. The routers in the path are not allowed to decapsulate the packets they receive unless they need to be fragmented.



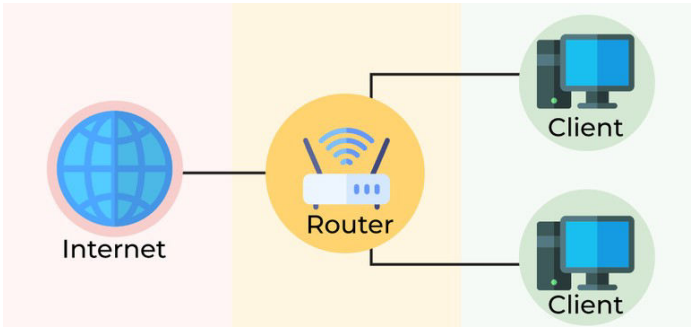
2. Routing

Routing is the process of moving data from one device to another device. These are two other services offered by the network layer. In a network, there are a number of routes available from the source to the destination. The network layer specifies some strategies which find out the best possible route. This process is referred to as routing. There are a number of routing protocols that are used in this process and they should be run to help the routers coordinate with each other and help in establishing communication throughout the network.



3. Forwarding

Forwarding is simply defined as the action applied by each router when a packet arrives at one of its interfaces. When a router receives a packet from one of its attached networks, it needs to forward the packet to another attached network (unicast routing) or to some attached networks (in the case of multicast routing). Routers are used on the network for forwarding a packet from the local network to the remote network. So, the process of routing involves packet forwarding from an entry interface out to an exit interface.



Difference between Routing and Forwarding

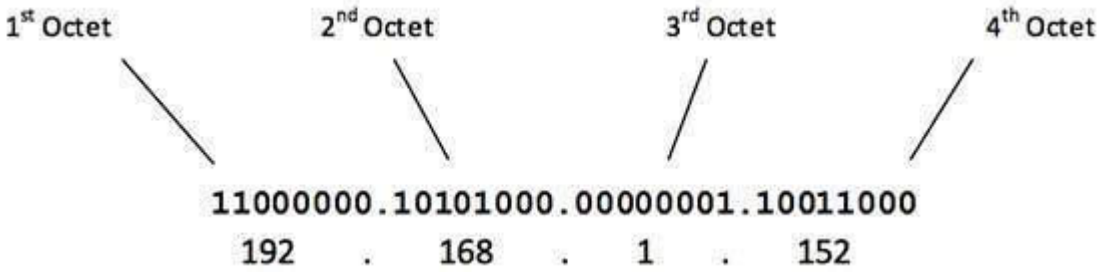
Routing	Forwarding
Routing is the process of moving data from one device to another device.	Forwarding is simply defined as the action applied by each router when a packet arrives at one of its interfaces.
Operates on the Network Layer.	Operates on the Network Layer.
Work is based on Forwarding Table.	Checks the forwarding table and work according to that.
Works on protocols like Routing Information Protocol (RIP) for Routing.	Works on protocols like UDP Encapsulating Security Payloads

IPv4 - Address Classes

Internet Protocol hierarchy contains several classes of IP Addresses to be used efficiently in various situations as per the requirement of hosts per network. Broadly, the IPv4 Addressing system is divided into five classes of IP Addresses. All the five classes are identified by the first octet of IP Address.

> Internet Corporation for Assigned Names and Numbers is responsible for assigning IP addresses.

The first octet referred here is the left most of all. The octets numbered as follows depicting dotted decimal notation of IP Address −



The number of networks and the number of hosts per class can be derived by this formula −

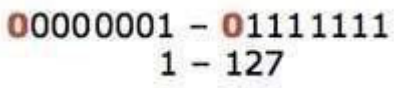
```
Number of networks          = 2^network_bits

Number of Hosts/Network     = 2^host_bits - 2
```

When calculating hosts' IP addresses, 2 IP addresses are decreased because they cannot be assigned to hosts, i.e. the first IP of a network is network number and the last IP is reserved for Broadcast IP.

Class A Address

The first bit of the first octet is always set to 0 (zero). Thus the first octet ranges from 1 – 127, i.e.



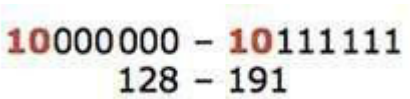
Class A addresses only include IP starting from 1.x.x.x to 126.x.x.x only. The IP range 127.x.x.x is reserved for loopback IP addresses.

The default subnet mask for Class A IP address is 255.0.0.0 which implies that Class A addressing can have 126 networks (2^7-2) and 16777214 hosts (2^{24}-2).

Class A IP address format is thus: **0NNNNNNN**.HHHHHHHH.HHHHHHHH.HHHHHHHH

Class B Address

An IP address which belongs to class B has the first two bits in the first octet set to 10, i.e.



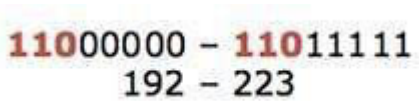
Class B IP Addresses range from 128.0.x.x to 191.255.x.x. The default subnet mask for Class B is 255.255.x.x.

Class B has 16384 (2^{14}) Network addresses and 65534 (2^{16}-2) Host addresses.

Class B IP address format is: **10NNNNNN.NNNNNNNN**.HHHHHHHH.HHHHHHHH

Class C Address

The first octet of Class C IP address has its first 3 bits set to 110, that is −



Class C IP addresses range from 192.0.0.x to 223.255.255.x. The default subnet mask for Class C is 255.255.255.x.

Class C gives 2097152 (2^{21}) Network addresses and 254 (2^8-2) Host addresses.

Class C IP address format is: **110NNNNN.NNNNNNNN.NNNNNNNN**.HHHHHHHH

Class D Address

Very first four bits of the first octet in Class D IP addresses are set to 1110, giving a range of −

Class D has IP address range from 224.0.0.0 to 239.255.255.255. Class D is reserved for Multicasting. In multicasting data is not destined for a particular host, that is why there is no need to extract host address from the IP address, and Class D does not have any subnet mask.

Class E Address

This IP Class is reserved for experimental purposes only for R&D or Study. IP addresses in this class ranges from 240.0.0.0 to 255.255.255.254. Like Class D, this class too is not equipped with any subnet mask.

IPV4 subnetting

Class A Subnets

In Class A, only the first octet is used as Network identifier and rest of three octets are used to be assigned to Hosts (i.e. 16777214 Hosts per Network). To make more subnet in Class A, bits from Host part are borrowed and the subnet mask is changed accordingly.

For example, if one MSB (Most Significant Bit) is borrowed from host bits of second octet and added to Network address, it creates two Subnets (2^1=2) with (2^{23}-2) 8388606 Hosts per Subnet.

Class B Subnets

By default, using Classful Networking, 14 bits are used as Network bits providing (2^{14}) 16384 Networks and (2^{16}-2) 65534 Hosts. Class B IP Addresses can be subnetted the same way as Class A addresses, by borrowing bits from Host bits

Class C Subnets

Class C IP addresses are normally assigned to a very small size network because it can only have 254 hosts in a network.

IPv6 Packet Format

An IPv6 packet has three parts: an IPv6 basic header, one or more IPv6 extension headers, and an upper-layer protocol data unit (PDU).

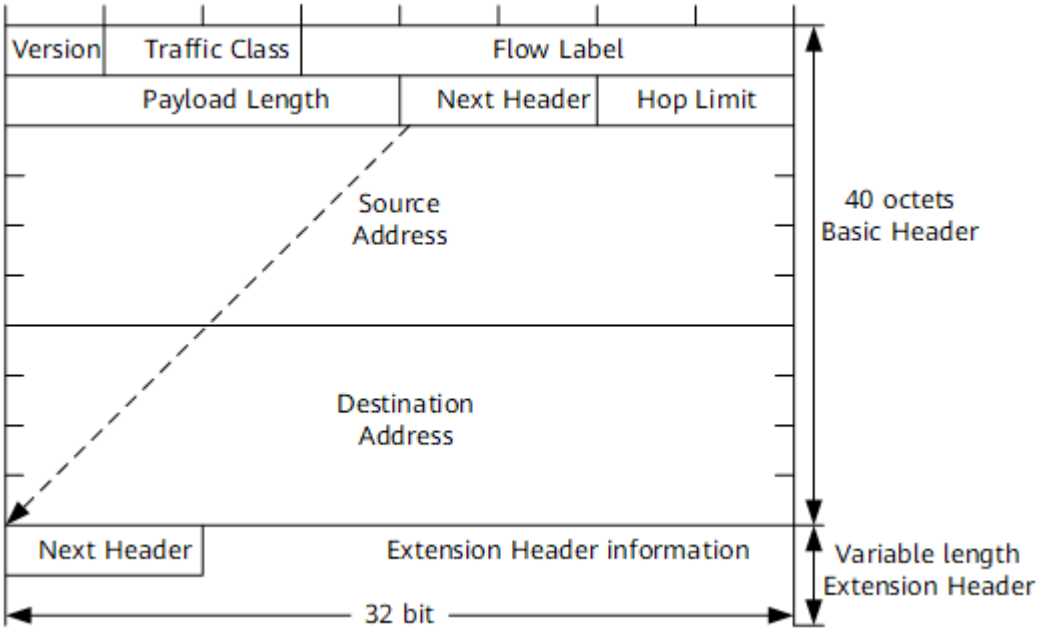
An upper-layer PDU is composed of the upper-layer protocol header and its payload, which maybe an ICMPv6 packet, a TCP packet, or a UDP packet.

IPv6 Basic Header

An IPv6 basic header is fixed as 40 bytes long and has eight fields. Each IPv6 packet must have an IPv6 basic header that provides basic packet forwarding information, and which all devices parse on the forwarding path.

Figure 11-7 shows the IPv6 basic header.

Figure 11-7 IPv6 basic header



An IPv6 basic header contains the following fields:

- Version: 4 bits long. In IPv6, the value of the Version field is set to 6.
- Traffic Class: 8 bits long. This field indicates the class or priority of an IPv6 packet. The Traffic Class field is similar to the TOS field in an IPv4 packet and is mainly used in QoS control.
- Flow Label: 20 bits long. This field was added in IPv6 to differentiate traffic. A flow label and source IP address identify a data flow. Intermediate network devices can effectively differentiate data flows based on this field.
- Payload Length: 16 bits long. This field indicates the length of the IPv6 payload in bytes. The payload is the part of the IPv6 packet following the IPv6 basic header, including the extension header and upper-layer PDU. This field has a maximum value of 65535. If the payload length exceeds 65535 bytes, the field is set to 0, and the Jumbo Payload option in the Hop-by-Hop Options header is used to express the actual payload length.
- Next Header: 8 bits long. This field identifies the type of the first extension header that follows the IPv6 basic header or the protocol type in the upper-layer PDU.
- Hop Limit: 8 bits long. This field is similar to the Time to Live field in an IPv4 packet, defining the maximum number of hops that an IP packet can pass through. Each device that forwards the packet decrements the field value by 1. If the field value is reduced to 0, the packet is discarded.
- Source Address: 128 bits long. This field indicates the address of the packet originator.
- Destination Address: 128 bits long. This field indicates the address of the packet recipient.

Unlike the IPv4 packet header, the IPv6 packet header does not carry IHL, identifier, flag, fragment offset, header checksum, option, or padding fields, but it carries the flow label field. This facilitates IPv6 packet processing and improves processing efficiency. To support various options without changing the existing packet format, the Extension Header information field is added to the IPv6 packet header, improving flexibility. The following paragraphs describe IPv6 extension headers.

Distance Vector Routing –

- It is a dynamic routing algorithm in which each router computes a distance between itself and each possible destination i.e. its immediate neighbors.
- The router shares its knowledge about the whole network to its neighbors and accordingly updates the table based on its neighbors.
- The sharing of information with the neighbors takes place at regular intervals.
  Problems – Count to infinity problem which can be solved by splitting horizon.
  – Good news spread fast and bad news spread slowly.
  – Persistent looping problem i.e. loop will be there forever.

Link State Routing –

- It is a dynamic routing algorithm in which each router shares knowledge of its neighbors with every other router in the network.
- A router sends its information about its neighbors only to all the routers through flooding.
- Information sharing takes place only whenever there is a change.
- **Problems –** Heavy traffic due to flooding of packets.
  – Flooding can result in infinite looping which can be solved by using the **Time to live (TTL)** field.

Differences

S.No.	Distance Vector Routing	Link State Routing
1.	Bandwidth required is less due to local sharing, small packets and no flooding.	Bandwidth required is more due to flooding and sending of large link state packets.
2.	Based on local knowledge, since it updates table based on information from neighbours.	Based on global knowledge, it have knowledge about entire network.
3	Traffic is less.	Traffic is more.
4.	Converges slowly i.e, good news spread fast and bad news spread slowly.	Converges faster.
5.	Count of infinity problem.	No count of infinity problem.
6	Persistent looping problem i.e, loop will be there forever.	No persistent loops, only transient loops.

Responsibilities of a Transport Layer

- The Process to Process Delivery
- End-to-End Connection between Hosts
- Multiplexing and Demultiplexing
- Congestion Control

- Data integrity and Error correction
- Flow control

1. The Process to Process Delivery

While Data Link Layer requires the MAC address (48 bits address contained inside the Network Interface Card of every host machine) of source-destination hosts to correctly deliver a frame and the Network layer requires the IP address for appropriate routing of packets, in a similar way Transport Layer requires a Port number to correctly deliver the segments of data to the correct process amongst the multiple processes running on a particular host. A port number is a 16-bit address used to identify any client-server program uniquely.

2. End-to-end Connection between Hosts

The transport layer is also responsible for creating the end-to-end Connection between hosts for which it mainly uses TCP and UDP.TCP is a secure, connection-orientated protocol that uses a handshake protocol to establish a robust connection between two end hosts. TCP ensures the reliable delivery of messages and is used in various applications. UDP, on the other hand, is a stateless and unreliable protocol that ensures best-effort delivery. It is suitable for applications that have little concern with flow or error control and requires sending the bulk of data like video conferencing. It is often used in multicasting protocols.

3. Multiplexing and Demultiplexing

Multiplexing(many to one) is when data is acquired from several processes from the sender and merged into one packet along with headers and sent as a single packet. Multiplexing allows the simultaneous use of different processes over a network that is running on a host. The processes are differentiated by their port numbers. Similarly, Demultiplexing(one to many is required at the receiver side when the message is distributed into different processes. Transport receives the segments of data from the network layer distributes and delivers it to the appropriate process running on the receiver's machine.

4. Congestion Control

Congestion is a situation in which too many sources over a network attempt to send data and the router buffers start overflowing due to which loss of packets occurs. As a result, the retransmission of packets from the sources increases the congestion further. In this situation, the Transport layer provides Congestion Control in different ways. It uses open-loop congestion control to prevent congestion and closed-loop congestion control to remove the congestion in a network once it occurred. TCP provides AIMD – additive increases multiplicative decrease and leaky bucket technique for congestion control.

Leaky bucket technique : Suppose we have a bucket in which we are pouring water, at random points in time, but we have to get water at a fixed rate, to achieve this we will make a hole at the bottom of the bucket. This will ensure that the water coming out is at some fixed rate, and also if the bucket gets full, then we will stop pouring water into it.

The input rate can vary, but the output rate remains constant. Similarly, in networking, a technique called leaky bucket can smooth out bursty traffic. Bursty chunks are stored in the bucket and sent out at an average rate.

5. Data integrity and Error Correction

The transport layer checks for errors in the messages coming from the application layer by using error detection codes, and computing checksums, it checks whether the received data is not corrupted and uses the ACK and NACK services to inform the sender if the data has arrived or not and checks for the integrity of data.

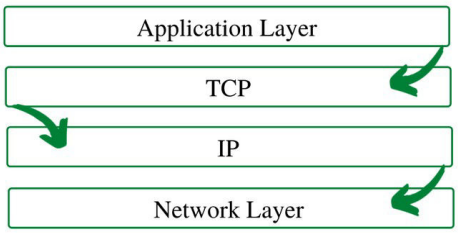
6. Flow Control

The transport layer provides a flow control mechanism between the adjacent layers of the TCP/IP model. TCP also prevents data loss due to a fast sender and slow receiver by imposing some flow control techniques. It uses the method of sliding window protocol which is accomplished by the receiver by sending a window back to the sender informing the size of data it can receive.

Protocols of Transport Layer

TCP

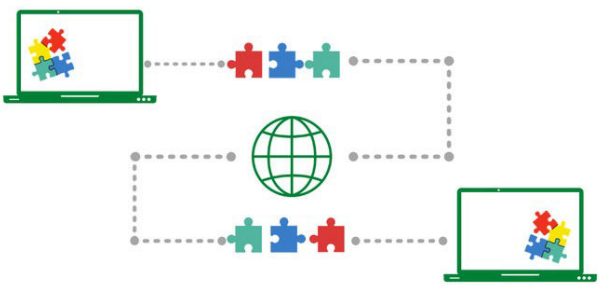
TCP (Transmission Control Protocol) is one of the main protocols of the Internet protocol suite. It lies between the Application and Network Layers which are used in providing reliable delivery services. It is a connection-oriented protocol for communications that helps in the exchange of messages between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



Working of TCP

To make sure that each message reaches its target location intact, the TCP/IP model breaks down the data into small bundles and afterward reassembles the bundles into the original message on the opposite end. Sending the information in little bundles of information makes it simpler to maintain efficiency as opposed to sending everything in one go.

After a particular message is broken down into bundles, these bundles may travel along multiple routes if one route is jammed but the destination remains the same.



We can see that the message is being broken down, then reassembled from a different order at the destination

For example, When a user requests a web page on the internet, somewhere in the world, the server processes that request and sends back an HTML Page to that user. The server makes use of a protocol called the HTTP Protocol. The HTTP then requests the TCP layer to set the required connection and send the HTML file.

Now, the TCP breaks the data into small packets and forwards it toward the Internet Protocol (IP) layer. The packets are then sent to the destination through different routes.

The TCP layer in the user's system waits for the transmission to get finished and acknowledges once all packets have been received.

Features of TCP/IP

Some of the most prominent features of Transmission control protocol are

1. Segment Numbering System

- TCP keeps track of the segments being transmitted or received by assigning numbers to each and every single one of them.
- A specific *Byte Number* is assigned to data bytes that are to be transferred while segments are assigned *sequence numbers*.
- *Acknowledgment Numbers* are assigned to received segments.

2. Connection Oriented

- It means sender and receiver are connected to each other till the completion of the process.
- The order of the data is maintained i.e. order remains same before and after transmission.

3. Full Duplex

- In TCP data can be transmitted from receiver to the sender or vice – versa at the same time.
- It increases efficiency of data flow between sender and receiver.

4. Flow Control

- Flow control limits the rate at which a sender transfers data. This is done to ensure reliable delivery.
- The receiver continually hints to the sender on how much data can be received (using a sliding window)

5. Error Control

- TCP implements an error control mechanism for reliable data transfer
- Error control is byte-oriented
- Segments are checked for error detection
- Error Control includes – *Corrupted Segment & Lost Segment Management, Out-of-order segments, Duplicate segments*, etc.

6. Congestion Control

- TCP takes into account the level of congestion in the network
- Congestion level is determined by the amount of data sent by a sender

Advantages

- It is a reliable protocol.
- It provides an error-checking mechanism as well as one for recovery.
- It gives flow control.
- It makes sure that the data reaches the proper destination in the exact order that it was sent.
- Open Protocol, not owned by any organization or individual.
- It assigns an IP address to each computer on the network and a domain name to each site thus making each device site to be distinguishable over the network.

Disadvantages

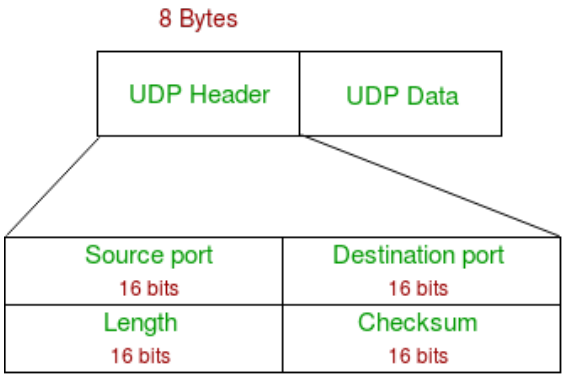
- TCP is made for Wide Area Networks, thus its size can become an issue for small networks with low resources.
- TCP runs several layers so it can slow down the speed of the network.
- It is not generic in nature. Meaning, it cannot represent
- any protocol stack other than the TCP/IP suite. E.g., it cannot work with a Bluetooth connection.
- No modifications since their development around 30 years ago.

User Datagram Protocol (UDP)

User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of the Internet Protocol suite, referred to as UDP/IP suite. Unlike TCP, it is an **unreliable and connectionless protocol.** So, there is no need to establish a connection prior to data transfer. The UDP helps to establish low-latency and loss-tolerating connections establish over the network.The UDP enables process to process communication.
Though Transmission Control Protocol (TCP) is the dominant transport layer protocol used with most of the Internet services; provides assured delivery, reliability, and much more but all these services cost us additional overhead and latency. Here, UDP comes into the picture. For real-time services like computer gaming, voice or video communication, live conferences; we need UDP. Since high performance is needed, UDP permits packets to be dropped instead of processing delayed packets. There is no error checking in UDP, so it also saves bandwidth.
User Datagram Protocol (UDP) is more efficient in terms of both latency and bandwidth.

UDP Header –

UDP header is an **8-bytes** fixed and simple header, while for TCP it may vary from 20 bytes to 60 bytes. The first 8 Bytes contains all necessary header information and the remaining part consist 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.



1. **Source Port:** Source Port is a 2 Byte long field used to identify the port number of the source.
2. **Destination Port:** It is a 2 Byte long field, used to identify the port of the destined packet.
3. **Length:** Length is the length of UDP including the header and the data. It is a 16-bits field.
4. **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.
- UDP is widely used in online gaming, where low latency and high-speed communication is essential for a good gaming experience. Game servers often send small, frequent packets of data to clients, and UDP is well suited for this type of communication as it is fast and lightweight.
- Streaming media applications, such as IPTV, online radio, and video conferencing, use UDP to transmit real-time audio and video data. The loss of some packets can be tolerated in these applications, as the data is continuously flowing and does not require retransmission.
- 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.
- Actually, UDP is a null protocol if you remove the checksum field.
    - Reduce the requirement of computer resources.
    - When using the Multicast or Broadcast to transfer.
    - The transmission of Real-time packets, mainly in multimedia applications.

Advantages of UDP:

1. Speed: UDP is faster than TCP because it does not have the overhead of establishing a connection and ensuring reliable data delivery.

2. Lower latency: Since there is no connection establishment, there is lower latency and faster response time.

3. Simplicity: UDP has a simpler protocol design than TCP, making it easier to implement and manage.

4. Broadcast support: UDP supports broadcasting to multiple recipients, making it useful for applications such as video streaming and online gaming.

5. Smaller packet size: UDP uses smaller packet sizes than TCP, which can reduce network congestion and improve overall network performance.

Disadvantages of UDP:

1. No reliability: UDP does not guarantee delivery of packets or order of delivery, which can lead to missing or duplicate data.

2. No congestion control: UDP does not have congestion control, which means that it can send packets at a rate that can cause network congestion.

3. No flow control: UDP does not have flow control, which means that it can overwhelm the receiver with packets that it cannot handle.

4. 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.

5. 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.