UNIT-4

Transport Layer responsibilities

The transport Layer is the second layer in the TCP/IP model and the fourth layer in the OSI model. It is an end-to-end layer used to deliver messages to a host. It is termed an end-to-end layer because it provides a point-to-point connection rather than hop-to-hop, between the source host and destination host to deliver the services reliably. The unit of data encapsulation in the Transport Layer is a segment.

Working of Transport Layer

The transport layer takes services from the Application layer and provides services to the Network layer.

At the sender's side: The transport layer receives data (message) from the Application layer and then performs Segmentation, divides the actual message into segments, adds the source and destination's port numbers into the header of the segment, and transfers the message to the Network layer.

At the receiver's side: The transport layer receives data from the Network layer, reassembles the segmented data, reads its header, identifies the port number, and forwards the message to the appropriate port in the Application layer.

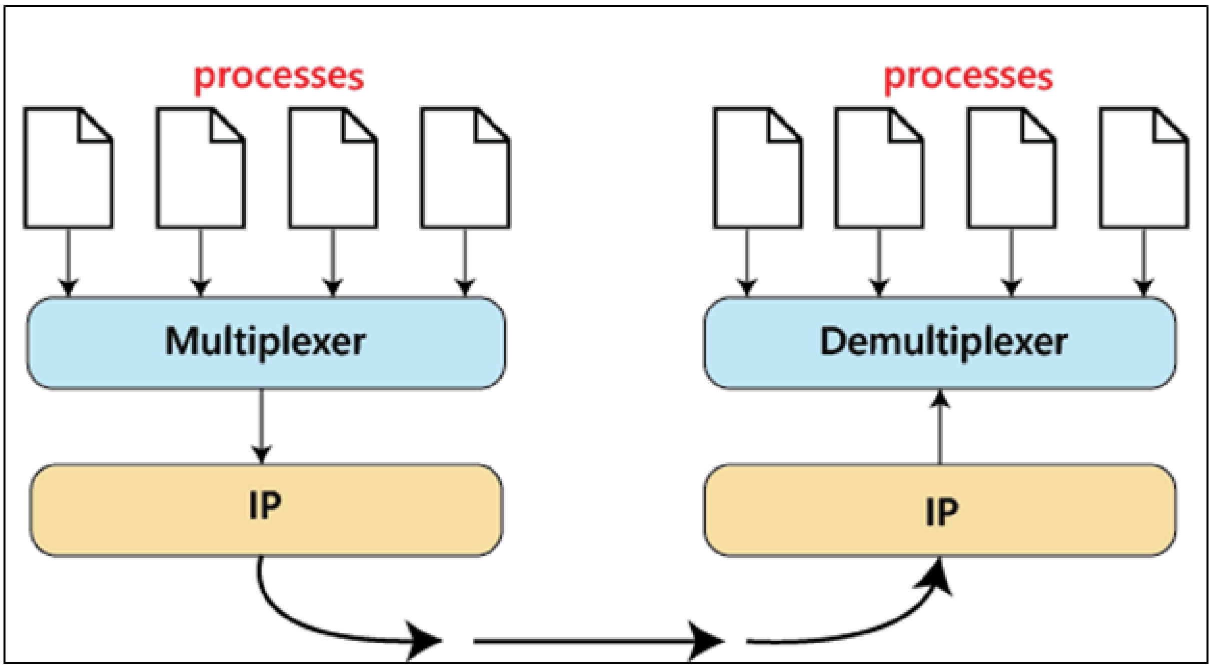
Responsibilities of a Transport Layer

- The Process to Process Delivery
- End-to-End Connection between Hosts (Client and Server)
- 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.

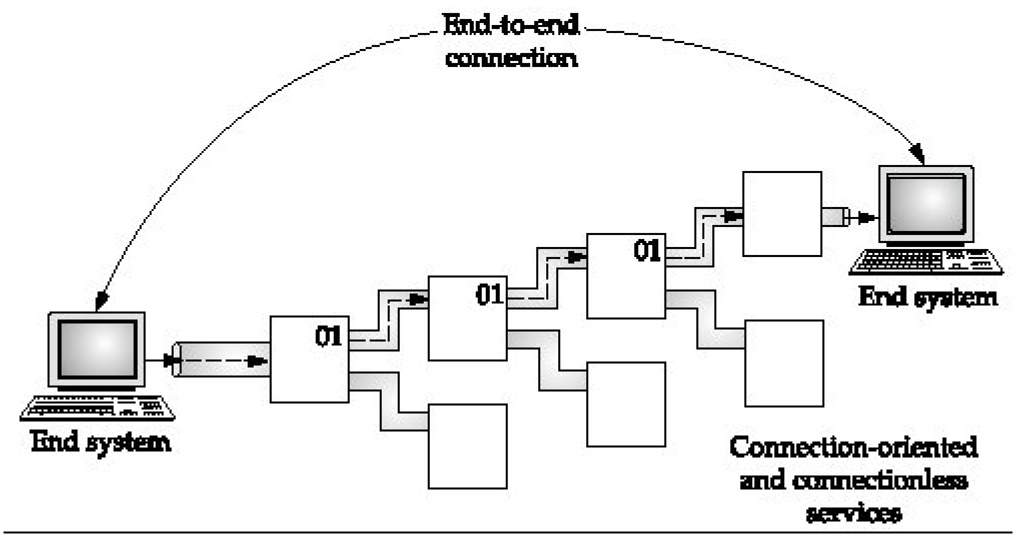
UNIT-4



Process to Process Delivery

2. End-to-end Connection between Hosts (Client and Server)

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.

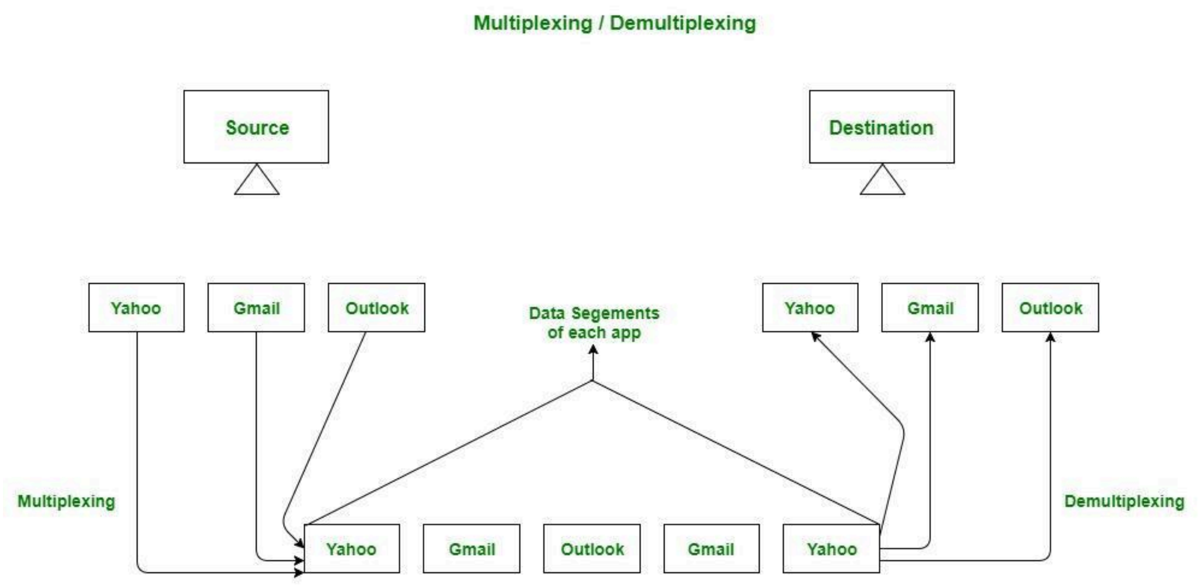


End to End Connection.

UNIT-4

3. Multiplexing and Demultiplexing

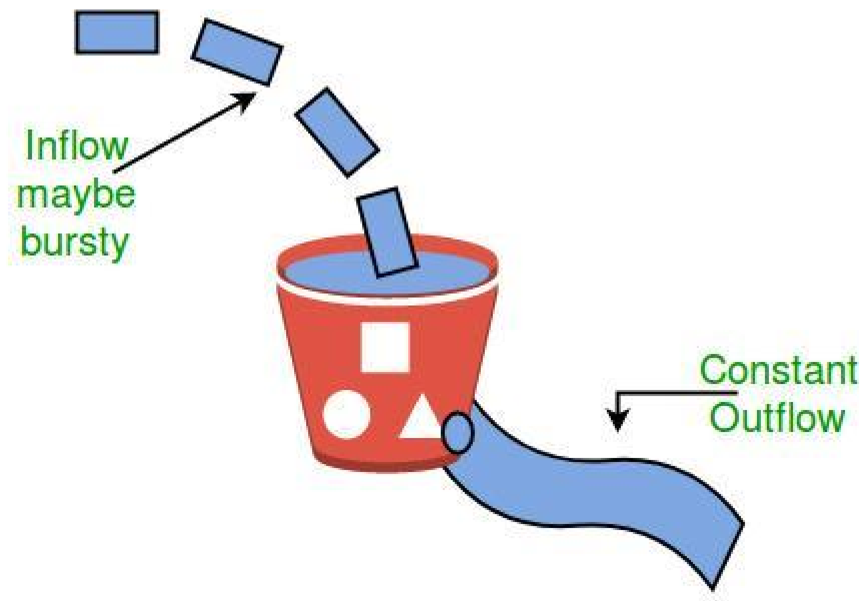
Multiplexing(many to one) is when data is acquired from several processes from the sender and merged into one packet along with headers and sent as a single packet. Multiplexing allows the simultaneous use of different processes over a network that is running on a host. The processes are differentiated 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.



Multiplexing and Demultiplexing

4. Congestion Control

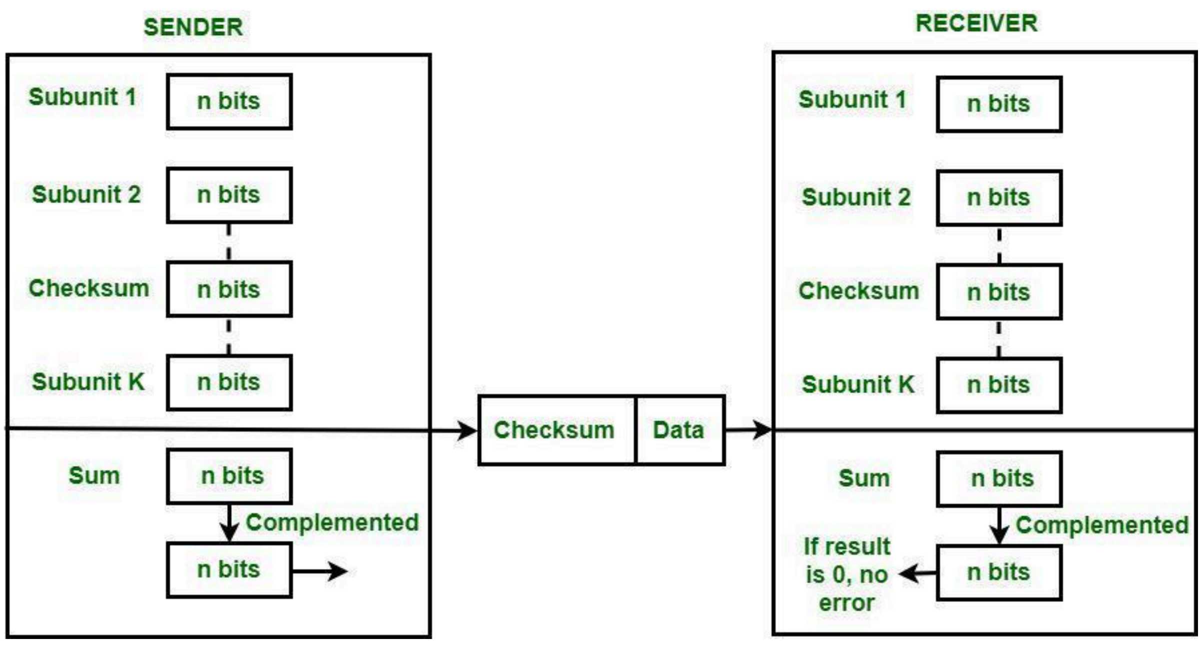
Congestion is a situation in which too many sources over a network attempt to send data and the router buffers start overflowing due to which loss of packets occurs. As a result, the retransmission of packets from the sources increases the congestion further. In this situation, the Transport layer provides 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 Congestion Control Technique

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.



Error Correction using Checksum

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

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- Stream Control Transmission Protocol (SCTP)

UNIT-4

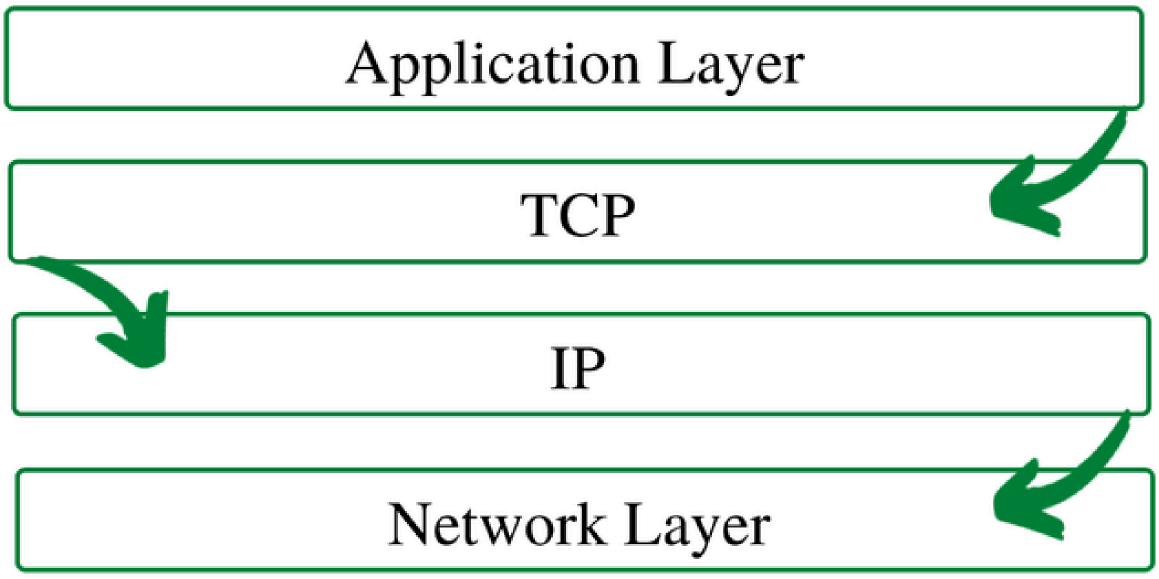
What is Transmission Control Protocol (TCP)?

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.

What is Transmission Control Protocol (TCP)?

Transmission Control Protocol 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.

The position of TCP is at the transport layer of the OSI model. TCP also helps in ensuring that information is transmitted accurately by establishing a virtual connection between the sender and receiver.



What is Internet Protocol (IP)?

Internet Protocol is a method that is useful for sending data from one device to another from all over the internet. Every device contains a unique IP Address that helps it communicate and exchange data across other devices present on the internet.

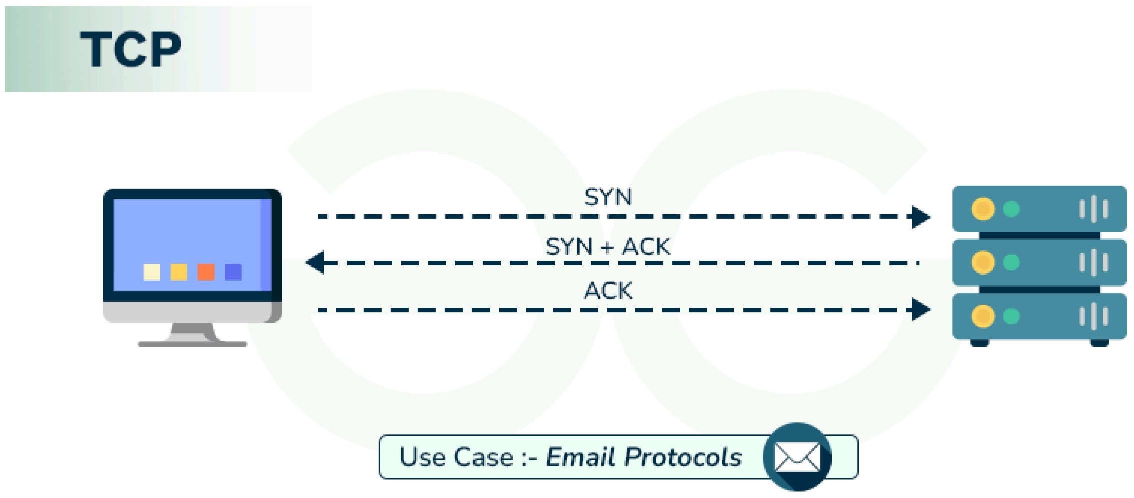
For more, you can refer to the TCP/IP Model.

Working of Transmission Control Protocol (TCP)

To make sure that each message reaches its target location undamaged, 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.

UNIT-4



TCP

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 mentioned below.

- **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.
- **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.
- **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.
- **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).
- **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.

UNIT-4

- **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 of TCP

- 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 unique over the network.

Disadvantages of TCP

- 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.

TCP Connection

TCP (Transmission Control Protocol) is a transmission protocol that ensures data transmission in an ordered and secure manner. It sends and receives the data packets in the same order. TCP is a **four-layer** protocol compared to OSI (Open System Interconnection Model), which is a **seven-layer** transmission process. It is recommended to transmit data from high-level protocols due to its integrity and security between the server and client.

To establish a connection, TCP needs a 3-way handshake. So, here we will discuss the detailed process of TCP to build a **3-way handshake** for connection. Here, we will discuss the following:

What is TCP?

TCP is a connection-oriented protocol, which means that it first establishes the connection between the sender and receiver in the form of a **handshake**. After both the connections are verified, it begins transmitting packets. It makes the transmission process error-free and ensures the delivery of data. It is an important part of the communication protocols used to interconnect network devices on the internet. The whole internet system relies on this network.

TCP is one of the most common protocols that ensure **end-to-end** delivery. It guarantees the security and integrity of the data being transmitted. It always establishes a secure connection between the sender and receiver. The transmitter is the **server**, and the receiver is known as the **client**. We can also say that the data transmission occurs between the server and client. Hence, TCP is used in most of the high-level protocols, such as **FTP** (File Transfer Protocol), **HTTP** (Hyper Text Transfer Protocol), and **SMTP** (Simple Mai Transfer Protocol).

Layers of TCP

The data is then divided into packets, assigned to the address, transmitted, routed, and received at the destination. The transmission process comprises four layers, application layer, transport layer, internet layer, and data link layer. The **application layer** performs the function similar to the top three layers (application, presentation, and session) of the OSI model and control user-interface specifications. The user interacts with the application layer of the TCP model, such as messaging and email systems. The **transport layer** provides a reliable and error-free data connection. It divides the data received from the application layer into packets, which helps in creating ordered sequence. The **internet layer** controls the routing of packet and ensures the delivery of a packet at the destination. The data link layer performs the function similar to the bottom two layers (data link and physical) of the OSI model. It is responsible for transmitting the data between the applications or devices in the network.

TCP Connection (A 3-way handshake)

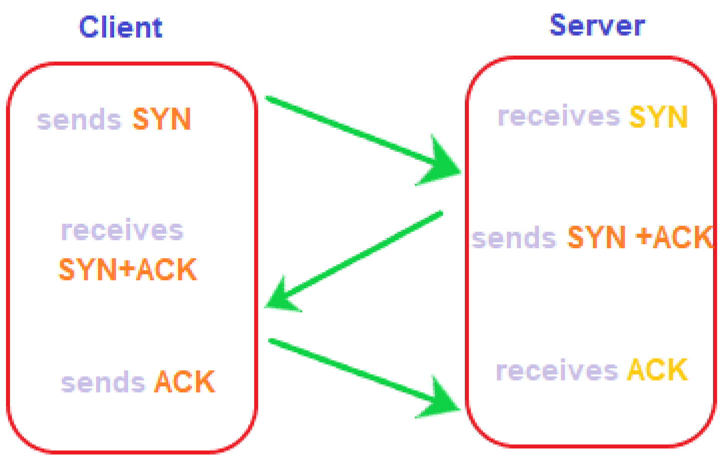
Handshake refers to the process to establish connection between the client and server. Handshake is simply defined as the process to establish a communication link. To transmit a packet, TCP needs a three way handshake before it starts sending data. The reliable communication in TCP is termed as **PAR** (Positive Acknowledgement Re-transmission).

When a sender sends the data to the receiver, it requires a positive acknowledgement from the receiver confirming the arrival of data. If the acknowledgement has not reached the sender, it needs to resend that data. The positive acknowledgement from the receiver establishes a successful connection.

Here, the server is the server and client is the receiver. The below diagram shows 3 steps for successful connection. A 3-way handshake is commonly known as SYN-SYN-ACK and requires both the client and server response to exchange the data. SYN means **synchronize Sequence Number** and ACK means **acknowledgment**. Each step is a type of handshake between the sender and the receiver.

The diagram of a successful TCP connection showing the three handshakes is shown below:

UNIT-4



The three handshakes are discussed in the below steps:

Step 1: SYN

SYN is a segment sent by the client to the server. It acts as a **connection request** between the client and server. It informs the server that the client wants to establish a connection. Synchronizing sequence numbers also helps synchronize sequence numbers sent between any two devices, where the same SYN segment asks for the sequence number with the connection request.

Step 2: SYN-ACK

It is an SYN-ACK segment or an SYN + ACK segment sent by the server. The ACK segment informs the client that the server has received the connection request and it is ready to build the connection. The SYN segment informs the sequence number with which the server is ready to start with the segments.

Step 3: ACK

ACK (Acknowledgment) is the last step before establishing a successful TCP connection between the client and server. The ACK segment is sent by the client as the response of the received ACK and SYN from the server. It results in the establishment of a reliable data connection.

After these three steps, the client and server are ready for the data communication process. TCP connection and termination are full-duplex, which means that the data can travel in both the directions simultaneously.

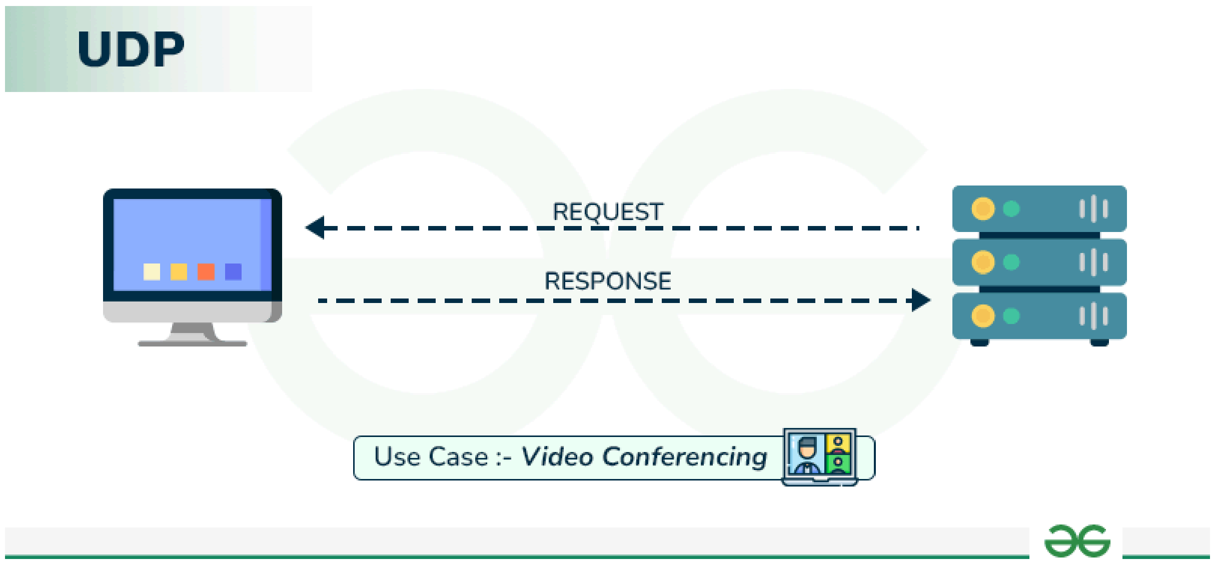
UNIT-4

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 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?

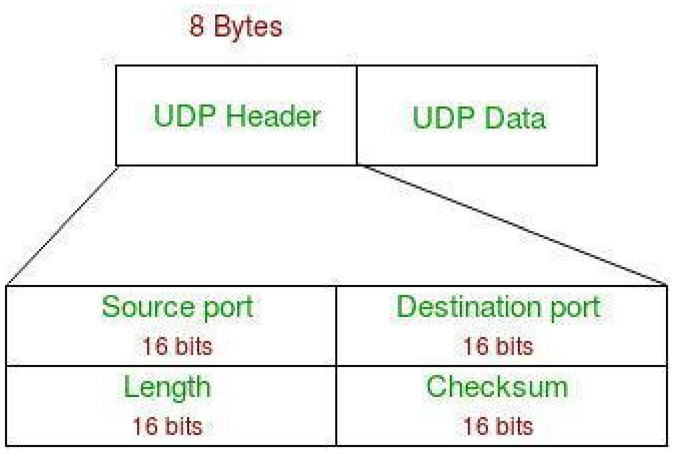
Though Transmission Control Protocol (TCP) is the dominant transport layer protocol used with most of the Internet services; provides assured delivery, reliability, and much more but all these services cost us additional overhead and latency. Here, UDP comes into the picture. For real-time services like computer gaming, voice or video communication, and 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.



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.

UNIT-4



UDP Header

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.

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.
- 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)
    - NNP (Network News Protocol)
    - Quote of the day protocol
- 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

- **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.

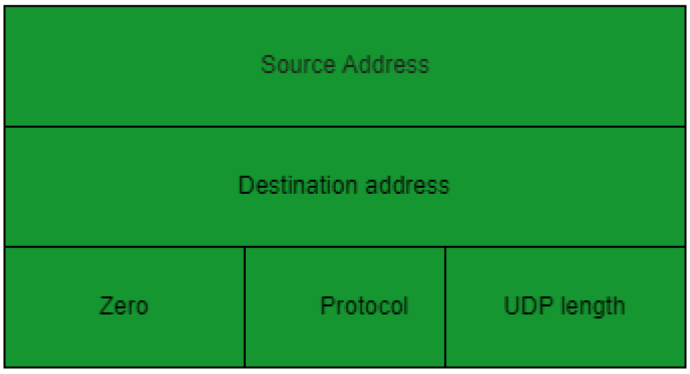
UNIT-4

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.
- **No flow control:** UDP does not have flow control, which means that it can overwhelm the receiver with packets that it cannot handle.
- **Vulnerable to attacks:** UDP is vulnerable to denial-of-service attacks, where an attacker can flood a network with UDP packets, overwhelming the network and causing it to crash.
- **Limited use cases:** UDP is not suitable for applications that require reliable data delivery, such as email or file transfers, and is better suited for applications that can tolerate some data loss, such as video streaming or online gaming.

UDP Pseudo Header

- The purpose of using a pseudo-header is to verify that the UDP packet has reached its correct destination
- The correct destination 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.
- The IP interface provides a level of abstraction that allows the UDP module to interact with the network layer without having to deal with the complexities of IP routing and addressing directly.
- The IP interface also handles fragmentation and reassembly of IP packets, which is important for large data transmissions that may exceed the maximum packet size allowed by the network.
- The IP interface may also provide additional services, such as support for Quality of Service (QoS) parameters and security mechanisms such as IPsec.
- The IP interface is a critical component of the Internet Protocol Suite, as it enables communication between hosts on the internet and allows for the seamless transmission of data packets across the network.

UNIT-4

SCTP Full Form

SCTP stands for **Stream Control Transmission Protocol**.

It is a connection- oriented protocol in computer networks which provides a full-duplex association i.e., transmitting multiple streams of data between two end points at the same time that have established a connection in network. It is sometimes referred to as next generation TCP, SCTP makes it easier to support telephonic conversation on Internet. A telephonic conversation requires transmitting of voice along with other data at the same time on both ends, SCTP protocol makes it easier to establish reliable connection.

SCTP is also intended to make it easier to establish connection over wireless network and managing transmission of multimedia data. SCTP is a standard protocol (RFC 2960) and is developed by Internet Engineering Task Force (IETF).

Characteristics of SCTP :

1. **Unicast with Multiple properties –**
   It is a point-to-point protocol which can use different paths to reach end host.
2. **Reliable Transmission –**
   It uses SACK and checksums to detect damaged, corrupted, discarded, duplicate and reordered data. It is similar to TCP but SCTP is more efficient when it comes to reordering of data.
3. **Message oriented –**
   Each message can be framed and we can keep order of datastream and tabs on structure. For this, In TCP, we need a different layer for abstraction.
4. **Multi-homing –**
   It can establish multiple connection paths between two end points and does not need to rely on IP layer for resilience.
5. **Security –**
   Another characteristic of SCTP that is security. In SCTP, resource allocation for association establishment only takes place following cookie exchange identification verification for the client (INIT ACK). Man-in-the-middle and denial-of-service attacks are less likely as a result. Furthermore, SCTP doesn’t allow for half-open connections, making it more resistant to network floods and masquerade attacks.

Advantages of SCTP :

1. It is a full- duplex connection i.e. users can send and receive data simultaneously.
2. It allows half- closed connections.
3. The message’s boundaries are maintained and application doesn’t have to split messages.
4. It has properties of both TCP and UDP protocol.
5. It doesn’t rely on IP layer for resilience of paths.

Disadvantages of SCTP :

1. One of key challenges is that it requires changes in transport stack on node.

2. Applications need to be modified to use SCTP instead of TCP/UDP.
3. Applications need to be modified to handle multiple simultaneous streams.

UNIT-4

Congestion Control in Computer Networks

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

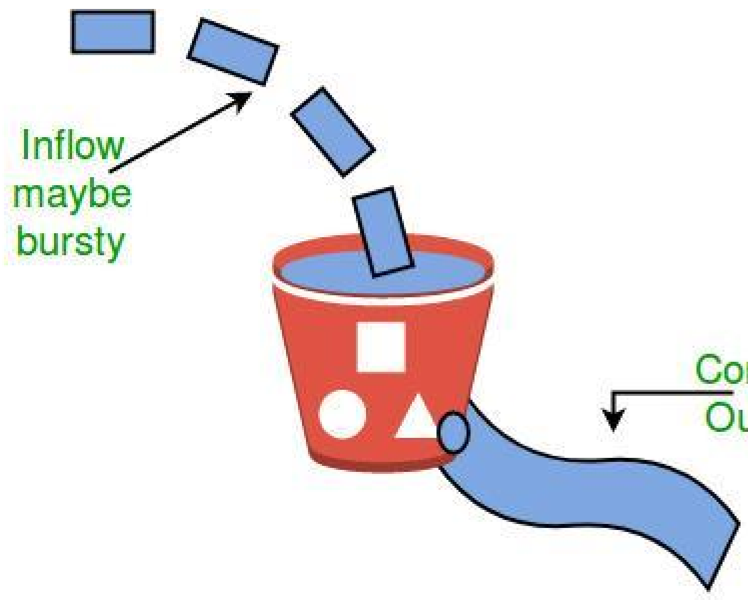
- Congestion Control is a mechanism that controls the entry of data packets into the network, enabling a better use of a shared network infrastructure and avoiding congestive collapse.
- Congestive-Avoidance Algorithms (CAA) are implemented at the TCP layer as the mechanism to avoid congestive collapse 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.

UNIT-4

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.

UNIT-4

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.

Steps of this algorithm can be described as follows:

1. In regular intervals tokens are thrown into the bucket.
2. The bucket has a maximum capacity.

3. If there is a ready packet, a token is removed from the bucket, and the packet is sent.
4. If there is no token in the bucket, the packet cannot be sent.

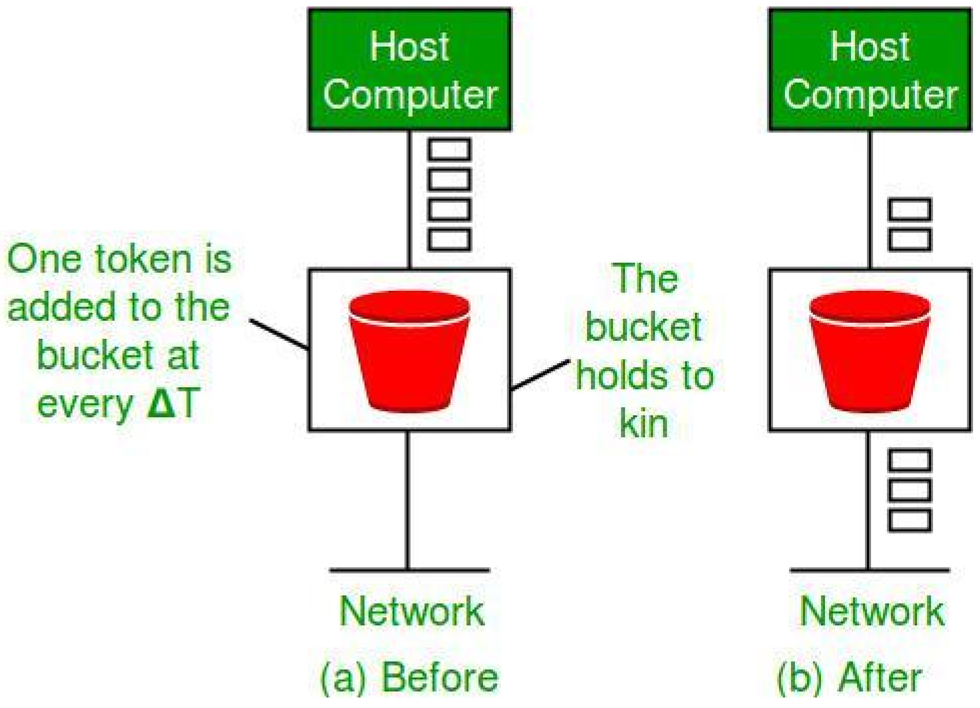
Let's understand with an example,

In figure (A) we see a bucket holding three tokens, with five packets waiting to be transmitted. For a packet to be transmitted, it must capture and destroy one token. In figure (B) We see that three of the five packets have gotten through, but the other two are stuck waiting for more tokens to be generated.

Ways in which token bucket is superior to leaky bucket: The leaky bucket algorithm controls the rate at which the packets are introduced in the network, but it is very conservative in nature. Some flexibility is introduced in the token bucket algorithm. In the token bucket, algorithm tokens are generated at each tick (up to a certain limit). For an incoming packet to be transmitted, it must capture a token and the transmission takes place at the same rate. Hence some of the busty packets are transmitted at the same rate if tokens are available and thus introduces some amount of flexibility in the system.

Formula: M * s = C + ? * s where S – is time taken M – Maximum output rate ? – Token arrival rate C – Capacity of the token bucket in byte

UNIT-4


Host Computer
Host Computer
One token is added to the bucket at every ΔT
The bucket holds to kin
Network
Network
(a) Before
(b) After

UNIT-4

Quality of Service

Quality-of-Service (QoS) refers to traffic control mechanisms that seek to either differentiate performance based on application or network-operator requirements or provide predictable or guaranteed performance to applications, sessions, or traffic aggregates. Basic phenomenon for QoS means in terms of packet delay and losses of various kinds.

Need for QoS –

- Video and audio conferencing require bounded delay and loss rate.
- Video and audio streaming requires bounded packet loss rate, it may not be so sensitive to delay.
- Time-critical applications (real-time control) in which bounded delay is considered to be an important factor.
- Valuable applications should be provided better services than less valuable applications.

QoS Specification –
QoS requirements can be specified as:

1. Delay
2. Delay Variation(Jitter)
3. Throughput
4. Error Rate

There are two types of QoS Solutions:

1. **Stateless Solutions –**
   Routers maintain no fine-grained state about traffic, one positive factor of it is that it is scalable and robust. But it has weak services as there is no guarantee about the kind of delay or performance in a particular application which we have to encounter.
2. **Stateful Solutions –**
   Routers maintain a per-flow state as flow is very important in providing the Quality-of-Service i.e. providing powerful services such as guaranteed services and high resource utilization, providing protection, and is much less scalable and robust.

Integrated Services(IntServ) –

1. An architecture for providing QoS guarantees in IP networks for individual application sessions.
2. Relies on resource reservation, and routers need to maintain state information of allocated resources and respond to new call setup requests.
3. Network decides whether to admit or deny a new call setup request.

IntServ QoS Components –

UNIT-4

- Resource reservation: call setup signaling, traffic, QoS declaration, per-element admission control.
- QoS-sensitive scheduling e.g WFQ queue discipline.
- QoS-sensitive routing algorithm(QSPF)
- QoS-sensitive packet discard strategy.

RSVP-Internet Signaling –
It creates and maintains distributed reservation state, initiated by the receiver and scales for multicast, which needs to be refreshed otherwise reservation times out as it is in soft state. Latest paths were discovered through "PATH" messages (forward direction) and used by RESV messages (reserve direction).

Call Admission –

- Session must first declare it's QoS requirement and characterize the traffic it will send through the network.
- **R-specification:** defines the QoS being requested, i.e. what kind of bound we want on the delay, what kind of packet loss is acceptable, etc.
- **T-specification:** defines the traffic characteristics like bustiness in the traffic.
- A signaling protocol is needed to carry the R-spec and T-spec to the routers where reservation is required.
- Routers will admit calls based on their R-spec, T-spec and based on the current resource allocated at the routers to other calls.

Diff-Serv –
Differentiated Service is a stateful solution in which each flow doesn't mean a different state. It provides reduced state services i.e. maintaining state only for larger granular flows rather than end-to-end flows tries to achieve the best of both worlds.

Intended to address the following difficulties with IntServ and RSVP:

1. **Flexible Service Models:**
   IntServ has only two classes, want to provide more qualitative service classes: want to provide 'relative' service distinction.
2. **Simpler signaling:**
   Many applications and users may only want to specify a more qualitative notion of service.

Streaming Live Multimedia –

- **Examples:** Internet radio talk show, Live sporting event.
- **Streaming:** playback buffer, playback buffer can lag tens of seconds after and still have timing constraint.
- **Interactivity:** fast forward is impossible, but rewind and pause is possible.