UNIT-4

**MODULE IV:** Transport Layer Connection Oriented and Connectionless Protocols -Process to Process Delivery, UDP and TCP protocols, SCTP. Congestion Control - Data Traffic, Congestion, Congestion Control, QoS, Integrated Services, Differentiated Services, QoS in Switched Networks.

**Transport Layer**

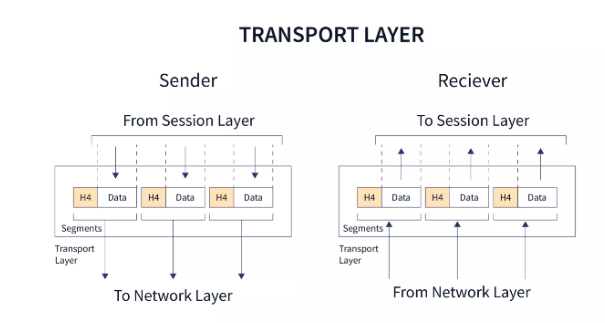
Transport Layer is the second layer in the [TCP/IP model](https://www.geeksforgeeks.org/tcp-ip-model/) and the fourth layer in the [OSI model](https://www.geeksforgeeks.org/layers-of-osi-model/). It is an end-to-end layer used to deliver messages to a host. It is termed an end-to-end layer because it 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 [Network layer](https://www.geeksforgeeks.org/network-layer-services-packetizing-routing-and-forwarding/) and provides services to the [Application layer](https://www.geeksforgeeks.org/application-layer-in-osi-model/)

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 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:**

**Process to process delivery:**

While Data Link Layer requires the [MAC address](https://www.geeksforgeeks.org/introduction-of-mac-address-in-computer-network/) (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.

**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 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.

**Multiplexing and Demultiplexing:**

Multiplexing allows simultaneous use of different applications over a network that is running on a host. The transport layer provides this mechanism which enables us to send packet streams from various applications simultaneously over a network. The transport layer accepts these packets from different processes differentiated by their port numbers and passes them to the network layer after adding proper headers. Similarly, Demultiplexing is required at the receiver side to obtain the data coming from various processes. Transport receives the segments of data from the network layer and delivers it to the appropriate process running on the receiver’s machine.

**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 occur. As a result 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 the congestion and closed-loop congestion control to remove the congestion in a network once it occurred. TCP provides AIMD- additive increase multiplicative decrease, leaky bucket technique for congestion control.

**Data integrity and Error correction:**

The transport layer checks for errors in the messages coming from the application layer by using error detection codes, 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.

**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.

## Transport Layer Protocols

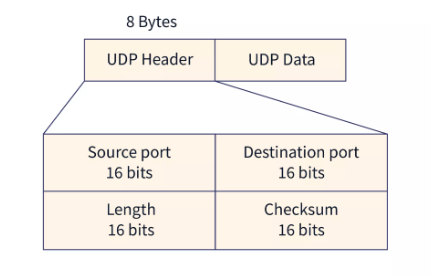
* UDP (User Datagram Protocol)
* TCP(Transmission Control Protocol)
* SCTP (Stream Control Transmission Protocol)

## UDP

* Connection less protocol
* Unreliable protocol
* UDP stands for User Datagram Protocol.
* UDP is one of the simplest transport layer protocol which provides non sequenced data transmission functionality.
* UDP is consider as connection less transport layer protocol.
* This type of protocol is referred to be used when speed and size are more important than reliability and security.
* It is an end-to-end transport level protocol that adds transport-level addresses, checksum error control, and length information to the data received from the upper layer.
* User datagram is the packet constructed by the UDP protocol

**Format of User Datagram**

Refer to the image below to see the header of UDP packet consisting of four fields.



User datagram have a fixed size header of 8 bytes which is divided into four parts -

**Source port address**: It defines source port number and it is of 16 bits.

**Destination port address:** It defines destination port number and it is of 16 bits.

**Total length:** This field is used to define the total length of the user datagram which is sum of header and data length in bytes. It is a 16-bit field.

**Checksum:** Checksum is also 16 bit field to carry the optional error detection data.

## UDP Services

* Process to Process Communication
* Connectionless Service
* Fast delivery of message
* Checksum

### Disadvantages

* UDP delivers basic functions required for the end-to-end transmission of data.
* It does not use any sequencing and does not identify the damaged packet while reporting an error.
* UDP can identify that an error has happened, but UDP does not identify which packet has been lost.

## TCP

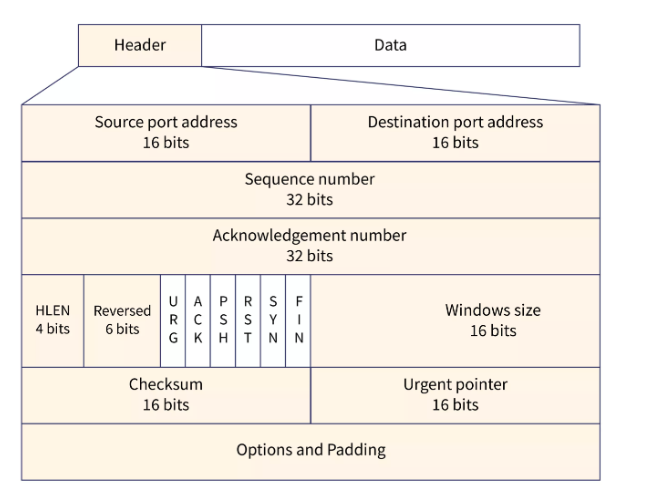
* Connection oriented protocol
* Reliable protocol
* Provide error and flow control
* TCP stands for Transmission Control Protocol.
* TCP is a connection-oriented transport layer protocol.
* TCP explicitly defines connection establishment, data transfer, and connection tear down phases to provide connection oriented service for data transmission.
* TCP is the most commonly used transport layer protocol.

### Features Of TCP protocol

* Stream data transfer
* Reliability
* Flow Control
* Error Control
* Multiplexing
* Logical Connections
* Full Duplex

## TCP Segment Format

Refer to the image below to see the header of TCP Segment.



* **Source port address** is a 16 bit field that defines port number of application program that is sending the segment.
* **Destination port address** is a 16 bit field that defines port number of application program that is receiving the segment.
* **Sequence number** is a field of 32 bit that will define the number assigned to data first byte contained in segment.
* **Acknowledgement number** is a 32 bit field that describe the next byte that receiver is looking forward to receive next from sender.
* **Header Length (HLEN)** is a field of 4 bit that specify the number of 4 byte words in TCP header. The header length of TCP header can be between 20 to 60 bytes.
* **Reserved** is a field 6 bit that are reserved for future use.
* **Control bits** are 6 different independent control bits or flags in this field.
* There are six in control field:
  1. URG: Urgent pointer
  2. ACK: Acknowledgement number
  3. PSH: Push request
  4. RST: Reset connection
  5. SYN: Sequence number Synchronization
  6. FIN: Connection termination
* **Window Size** is a 16-bit field that defines the size of the window of sending TCP in bytes.
* **Checksum**, 16-bit field contains checksum and used for error detection.
* **Urgent pointer** is a 16 bit field .This flag is set when there is urgent data in the data segment.
* **Options and padding** can be upto 40 bytes field for optional information in TCP header.

**SCTP: (Stream Control Transmission Protocol)**

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 or TCPng, 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.

**SCTP SERVICES**

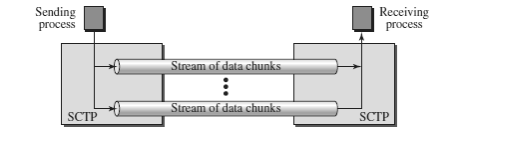
Process-to-Process Communication SCTP provides process-to-process communication.

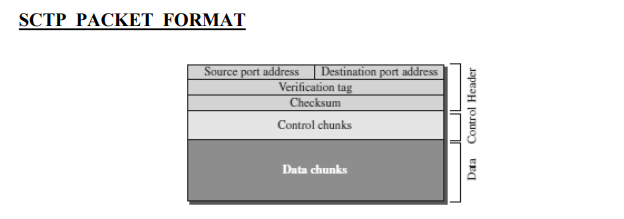
Multiple Streams

SCTP allows multistream service in each connection, which is called association in

SCTP terminology.

If one of the streams is blocked, the other streams can still deliver their data.





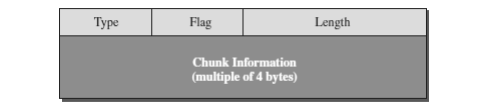
Chunks

Control information or user data are carried in chunks.

• Chunks have a common layout.

• The first three fields are common to all chunks; the information field depends on the

• type of chunk.



**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.

[](https://media.geeksforgeeks.org/wp-content/uploads/leaky.jpg)

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.

Steps of this algorithm can be described as follows:

1. In regular intervals tokens are thrown into the bucket. ƒ
2. The bucket has a maximum capacity. ƒ
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 + ρ \* s where S – is time taken M – Maximum output rate ρ – Token arrival rate C – Capacity of the token bucket in byte

Let’s understand with an example,

[](https://media.geeksforgeeks.org/wp-content/uploads/leakybuk.jpg)

**Quality-of-Service (QoS)** 

**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 –**

* 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.