EDU ENGINEERING

PIONEER OF ENGINEERING NOTES

TAMIL NADU'S BEST EDTECH PLATFORM FOR ENGINEERING

CONNECT WITH US

WEBSITE: www.eduengineering.net

TELEGRAM: @eduengineering

INSTAGRAM: @eduengineering

- **Regular Updates for all Semesters**
- **All Department** Notes AVAILABLE
- **Handwritten** Notes AVAILABLE
- **Past Year** Question Papers AVAILABLE
- **Subject wise** Question Banks AVAILABLE
- **Important Questions** for Semesters AVAILABLE
- Various **Author Books** AVAILABLE

Unit II: Transport Layer

Introduction - Transport Layer protocols: UDP - TCP: Connection management - Flow control - Congestion control - Congestion avoidance (DEC bit, RED) - SCTP - Quality of service

2.1 Introduction:

The transport layer is the heart of the TCP/IP protocol suite; it is the end to end logical vehicle for transferring data from one point to another in the Internet. It is responsible for end to end delivery of entire message.

Functions:

- This layer is the first one which breaks the messages into packets.
- This layer ensures that data must be received in the same sequence in which it was sent.
- This layer provides end to end delivery of data between hosts

2.1.1 Services:

Each protocol provides a different type of service and should be used appropriately

- UDP: UDP is an unreliable connectionless transport layer protocol used for its simplicity and efficiency.
- TCP: TCP is a reliable connection-oriented protocol that can be used in any application where reliability is important
- SCTP: New transport layer protocol that combines the features of UDP and TCP.

2.1.2 Port numbers :

Transport layer usually has several responsibilities. One is to create a process to process communication; these protocols use port numbers to accomplish this. Some of the well known ports used with UDP TCP is given below

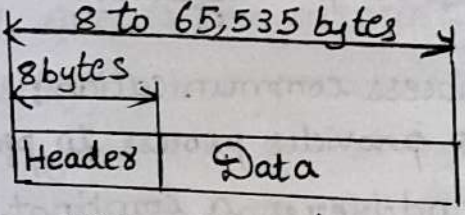
Port	Description	UDP	TCP	SCTP
7	Echoes back a received datagram	✓	✓	✓
9	Discards any datagram that is received	✓	✓	✓
11	Active users	✓	✓	✓
13	Returns the date and time	✓	✓	✓
20	File Transfer protocol	–	✓	✓
25	Simple Mail Transfer protocol	–	✓	✓
80	HyperText Transfer protocol	–	✓	✓

2.2 User Datagram Protocol

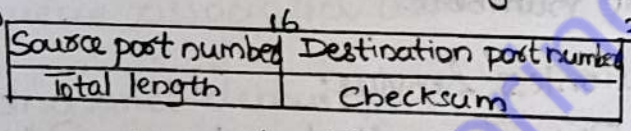
The User Datagram Protocol (UDP) is a connectionless, unreliable transport protocol. UDP is a very simple protocol using a minimum of overhead. If a process wants to send a small message and does not care much about reliability, It can use UDP. It also has no errors recovery procedures. It performs very limited error checking.

2.2.1 User Datagram

UDP packets called user datagrams, have a fixed size header of 8 bytes made of four fields, each of 2 bytes. Below figure shows the format of a user datagram.



a. UDP user data gram



b Header Format

Fig: Users Datagram Packet format

The fields are as follows:

- Source port number: This is the port number used by the process running on the source host. It is 16 bits long, which means that the port number can range from 0 to 65,535
- Destination port number: This is the port number used by the process running on the destination host. It is also 16 bits long.
- Length: This is a 16 bit field that defines the total length of the user datagram, header plus data
- Checksum: This field is used to detect errors

2.2.2 UDP Services:

Below are the general services provided by UDP

- Process to process communication
- connectionless services
- Flow control

- Error Control
- Checksum
- Congestion control
- Encapsulation and Decapsulation
- Queuing
- Multiplexing and Demultiplexing

- **Process to process communication:**

UDP provides process to process communication using socket addresses, a combination of IP addresses and port numbers.

- **Connectionless Services:**

This means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagram even if they are coming from the same source and going to the same destination.

- **Flow control:**

UDP is simple protocol. There is no flow control.

- **Error control:**

There is no error control mechanism in UDP except for the checksum. This means that the sender does not know if a message has been lost or duplicated.

- **Checksum:**

It is used by the sender and receiver to check for data corruption.

- **Congestion control:**

UDP does not provide congestion control UDP assumes that the packets sent are small and cannot create congestion in the network.

• Encapsulation and Decapsulation:

To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages.

• Queuing:

In UDP, queues are associated with ports. At the client site, when a process starts, it requests a port number from the operating system. Some implementations create both an incoming and an outgoing queue associated with each process.

• Multiplexing and demultiplexing:

In a host running a TCP/IP protocol suite, there is only one UDP but possibly several processes that may want to use the services of UDP. To handle this situation, UDP multiplexes and demultiplexes.

2.2.3 UDP Applications

1. UDP is suitable for a process that requires simple request-response communication.
2. UDP is suitable for a process with internal flow and error control mechanisms.
3. UDP is a suitable transport protocol for multicasting
4. UDP is used for management processes.
5. UDP is normally used for interactive real time applications that cannot tolerate uneven delay between sections of a received message.

2.3 Transmission Control Protocol

TCP is a connection oriented, reliable protocol. TCP is a process to process protocol. TCP uses flow and error control mechanism at the transport level.

2.3.1 TCP Services:

The services offered by TCP to the processes is explained below:

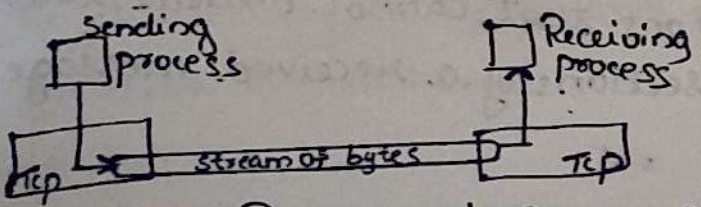
- Process to Process communication
- Stream Delivery services
- Full Duplex communication
- Multiplexing and Demultiplexing
- Connection oriented service
- Reliable service

- **Process to process communication:**

TCP provides process to process communication using port numbers.

- **Stream Delivery Services:**

TCP is stream oriented protocol. TCP allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes. TCP creates an environment in which the two processes seem to be connected by an imaginary tube that carries their data across the Internet



- **Sending and Receiving buffers:** Because the sending and the receiving processes may not write or read data at the same speed, TCP

needs buffers for storage. There are two buffers the sending buffer and the receiving buffer, one for each direction

■ Segments: At the transport Layer, TCP groups a number of bytes together into a packets called a segment.

• Full Duplex Communication:

TCP offers full duplex service, in which data can flow in both directions at the same time. Each TCP endpoints then has its own sending and receiving buffer and segments move in both directions.

• Multiplexing and Demultiplexing:

Like UDP, TCP peyorms multiplexing at the sender and demultiplexing at the receiver.

• Connection Oriented Service:

TCP is a connection-oriented protocol. When a process at site A wants to send and receive data from another process at site B, the following occurs:

1. The two TCPs establish a connection between them.
2. Data are exchanged in both directions
3. The connection is terminated.

• Reliable Service:

TCP is a reliable transport protocol. It uses an acknowledgement mechanism to check the safe and sound arrival of data.

2.3.2 Tcp features :

- **Numbering System:**

Although the Tcp software keeps track the segments being transmitted or received, their is no field for a segment number value in the segment header. Instead, there are two fields, called the sequence number and the acknowledgement number. These two field refer to a byte number and not a segment number.

- **Byte Number:** When Tcp received bytes of data from a process, Tcp stores them in the sending buffer and numbers them. The numbering does not necessarily start from 0. Instead, Tcp chooses an arbitrary number between 0 and $2^{32}-1$

- **Sequence Number:** After the bytes have been numbered, Tcp assigns a sequence number to each segment that is being sent. It is defined as follows:

1. The sequence number of the first segment is the ISN (initial sequence number) which is a random number.
2. The sequence number of any other segment is the sequence number of the previous segment plus the number of bytes carried by the previous segment.

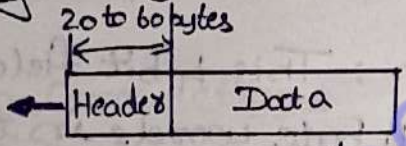
■ Acknowledgment Number:

Both sender and receiver uses an acknowledgment number to confirm the bytes it has received. The acknowledgment number defines the number of the next byte that the party expects to receive.

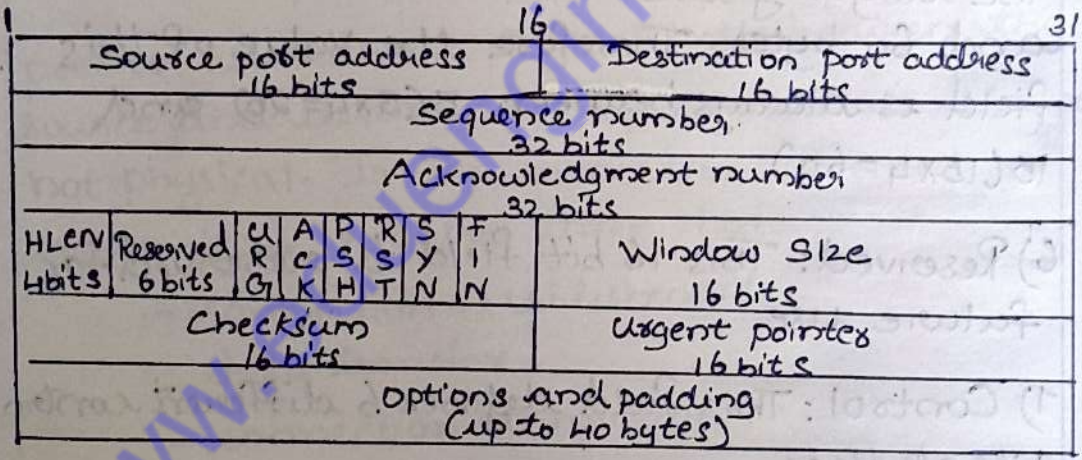
2.3.3 Segment

A packet in TCP is called a segment.

The format of a segment is shown in figure:



a. segment



b. Header

1) Source port address: This is a 16 bit field that defines the port number of the application program in the host that is sending the segment

2) Destination port address: This is a 16 bit field that defines the port number of the application program in the host that is receiving the segment.

3) Sequence number: This 32 bit field defines the number assigned to the first byte of data contained in this segment

4) Acknowledgment number: This 32 bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number x from other party, it returns x+1 as the acknowledgment number.

5) Header Length: This 4 bit field indicates the number of 4 byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore the value of this field is always between 5 (5×4=20) and 15 (15×4=60).

6) Reserved: This 16 bit field is reserved for future use

7) Control: This field defines 6 different control bits or flags

URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection (finish)

8) Window Size: This field defines the size of the window, in bytes, that the other party must maintain.

9) Checksum: This 16 bit field contains the checksum. The use of checksum in the UDP datagram is optional, whereas the use of the checksum for Tcp is mandatory.

10) Urgent pointer: This 16 bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data.

11) Options: There can be upto 40 bytes of optional information in the TCP header.

2.3.4. Tcp Connections

Tcp is a connection oriented transport protocol establishes a logical path between the source and destination. Tcp connection is logical not physical. In Tcp, connection-oriented transmission requires three phases:

- connection establishment
- data transfer
- connection termination

■ Connection establishment:

Tcp transmits data in full-duplex mode. When two TCPs in two machines are connected, they are able to send segments to each other simultaneously.

Three way Handshaking:

The connection establishment in Tcp is called three way handshaking. The three way handshaking process is shown below:

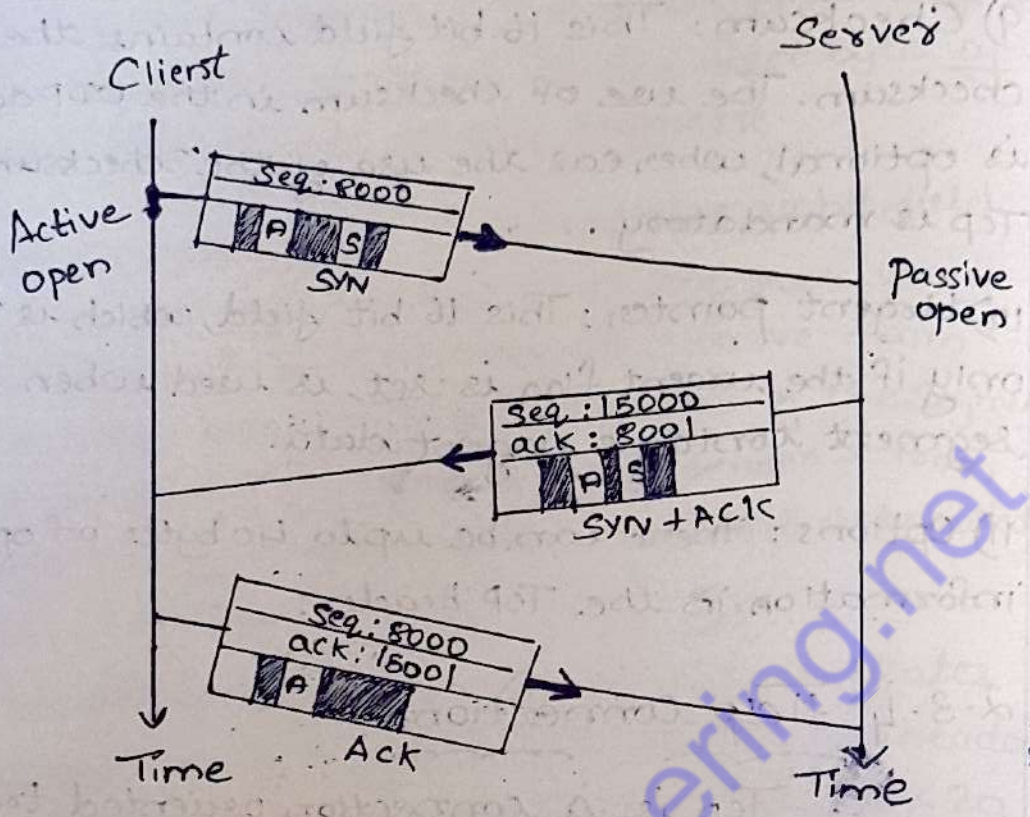


Fig: Connection establishment using three way handshaking

1. The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for synchronization of sequence numbers. It consumes one sequence number. When the data transfer start, the sequence number is incremented by 1.
2. The sever sends the second segment, a SYN+ACK segment, with 2 flag bits set: SYN and ACK.
3. The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field.

Data transfer :

After connection is established, bidirectional data transfer can take place. The client and server can both send data and acknowledge. Figure shows the example:

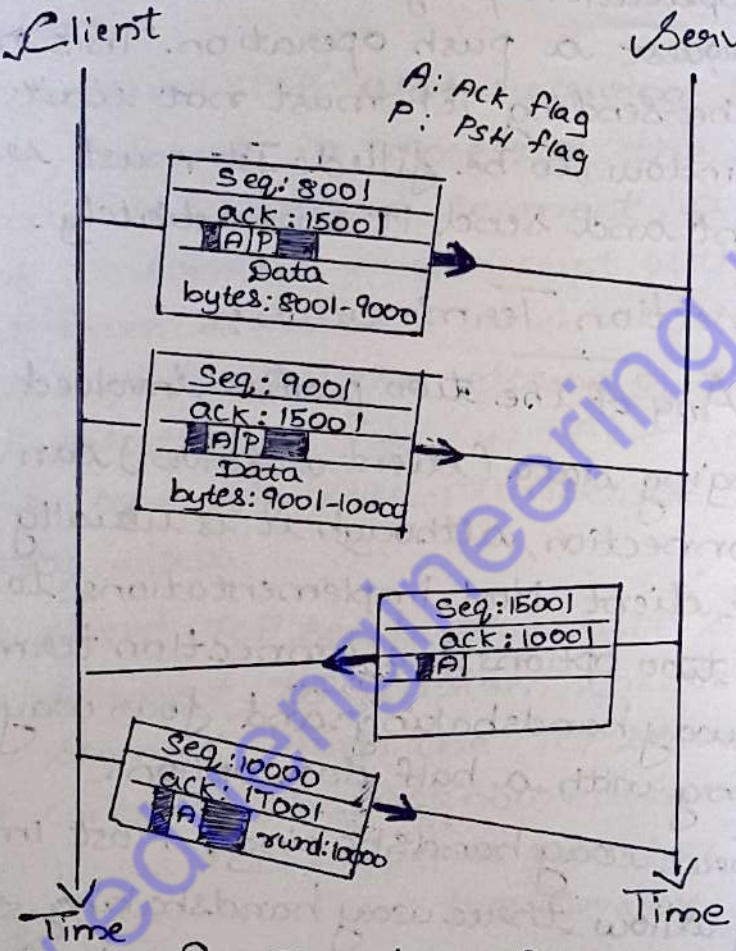


fig: Data transfer

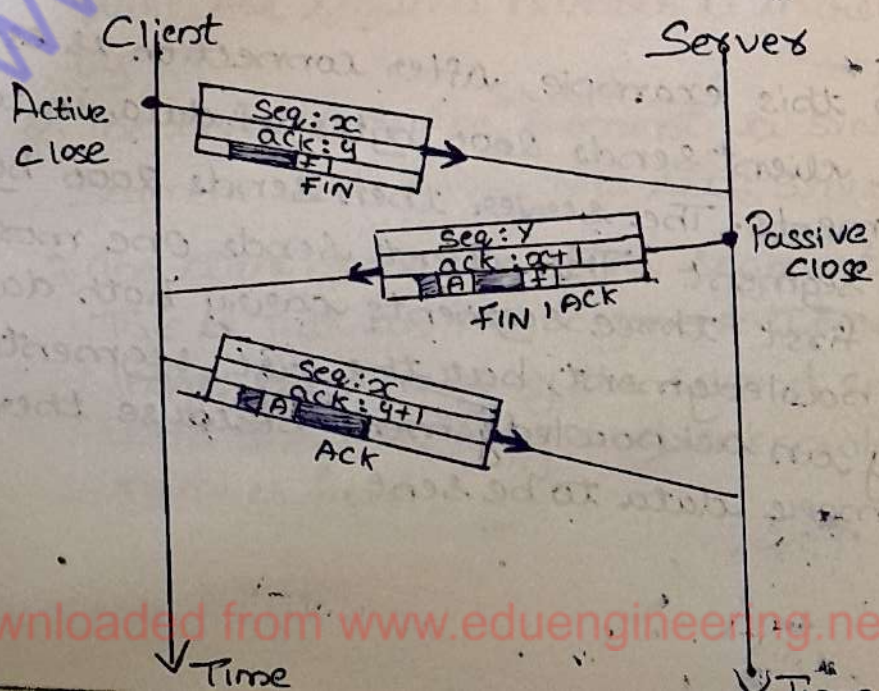
In this example, after connection is established the client sends 2000 bytes of data in two segments. The server then sends 2000 bytes in one segment. The client sends one more segment. The first three segments carry both data and acknowledgment, but the last segment carries only an acknowledgement because there are no more data to be sent.

Pushing data: The sending TCP uses a buffer to store the stream of data coming from the sending application program. The sending TCP can select the segment size. The application program at the sending site can request a push operation. This means that the sending TCP must not wait for the window to be filled. It must create a segment and send it immediately.

▪ Connection Termination:

Any of the two parities involved in exchanging data (client or server) can close the connection, although it is usually initiated by the client. Most implementations today allow two options for connection termination: three way handshaking and four way handshaking with a half close option.

- Three way handshaking: Most implementations today allow three way handshaking for connection termination as shown in figure



1. In a normal situation, the client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set. The FIN segment consumes one sequence number if it does not carry data

2. The Server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN + ACK segment, to confirm the receipt of the FIN Segment from the client and at the same time to announce the closing of the connection in other direction. The FIN + ACK segment consumes one sequence number if it does not carry data.

3. The client TCP sends the last segment an ACK segment, to confirm the receipt to the FIN Segment from the TCP Server. This Segment contains the acknowledgment number, which is 1 plus the sequence number received in the FIN segment from the Server.

2.3.5 Flow Control

Tcp uses a sliding window to handle flow control. The sliding window protocol used by TCP, however, is something between Go-Back-N and selective Repeat sliding window

The size of the window at one end is determined by the lesser of two values: receiver window (rwnd) or congestion window (cwnd)

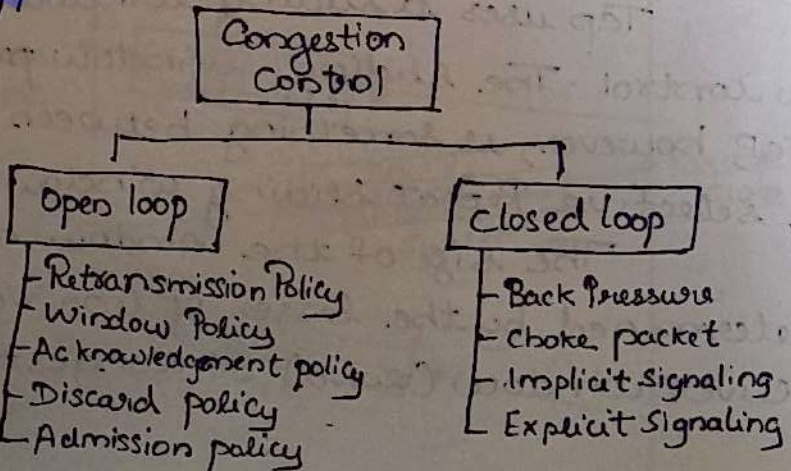
2.3.6 Error Control

TCP is a reliable transport layer protocol. This means that an application program that delivers a stream of data to TCP relies on TCP to deliver the entire stream to the application program on the other end in order, without error and without any part lost or duplicated

- Checksum: Each segment includes a checksum field which is used to check for a corrupted segment
- Acknowledgement: TCP uses acknowledgment to confirm the receipt of data segments.
- Retransmission: The heart of the error control mechanism is the retransmission of segments.

2.4 Congestion Control

Congestion control refers to techniques and mechanisms that can either prevent congestion before it happens or remove congestion, after it has happened.



In general, we can divide congestion control mechanism into two broad categories

- Open loop congestion control (Prevention)
- closed loop congestion control (removal)

■ Open loop congestion control:

In open loop congestion control, policies are applied to prevent congestion before it happens In these mechanisms, congestion control is handled by either the source or the destination.

1) Retransmission Policy: Retransmission is sometimes unavoidable. If the sender feels that a send packet is lost or corrupted, the packet needs to be retransmitted. Retransmission in general may increase congestion in the network. However a good retransmission policy can prevent congestion.

2) Window policy: The type of window at the sender may also affect congestion. The Selective Repeat window is better than the Go-Back-N window for congestion control. In the Go-Back-N window, when the timer for a packet times out, several packets may be resent

3) Acknowledgment Policy: The acknowledgment policy imposed by the receiver may also affect congestion. If the receiver does not acknowledge every packet it receives, it may slow down the sender and help prevent congestion.

4) Discarding policy: A good discarding policy by the routers may prevent congestion and at the same time may not harm the integrity of the transmission.

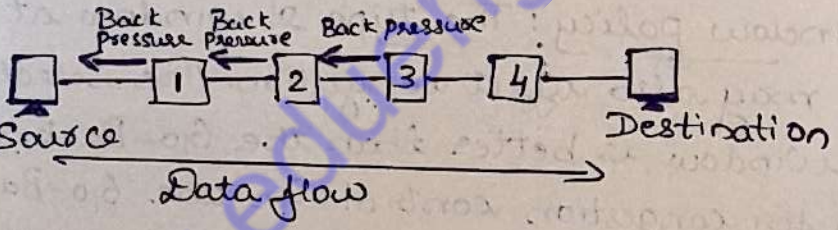
5) Admission Policy:

An admission policy, which is a quality of service mechanism, can also prevent congestion in virtual circuit networks.

Closed Loop Congestion Control

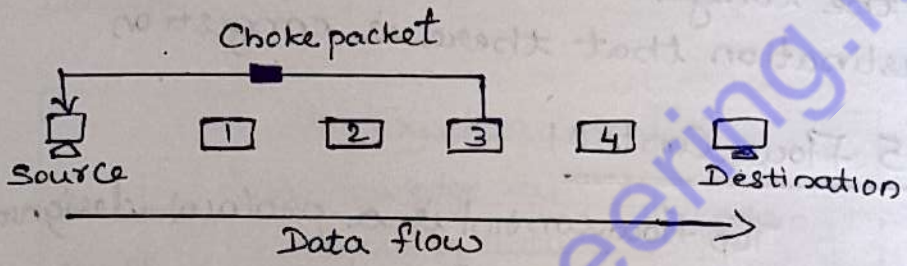
Closed loop congestion control mechanisms try to remove congestion after it occurs. Several mechanisms have been used by different protocols.

1) Backpressure: Backpressure is a node to node congestion control that start with a node and propagates, in the opposite direction of data flow, to the source. Figure shows the idea of back pressure.



Node 3 in the figure has more input data than it can handle. It drops some packets in its input buffer and informs node 11 to slow down. Node 11, in turn, may be congested because it is slowing down the output flow of data. If node 2 is congested, it informs to node 1 to slow down which in turn may create congestion. If so, node 1 inform the source of data to slow down. This, in time, alleviates the congestion. The pressure on node 3 is moved backward to source to remove the congestion.

2) Choke Packet : A choke packet is a packet sent by a node to the source to inform it of congestion. In backpressure, the warning is from one node to its upstream node to reach the source station. But in choke packet method, the warning is from the router, which has encountered congestion, to the source station directly.



3) Implicit Signaling : In implicit signaling, there is no communication between the congested node or nodes and the source. The source guesses that there is congestion somewhere in the network from other symptoms. For example, when a source sends several packets and there is no acknowledgment for a while, one assumption is that the network is congested and the source should slow down.

4) Explicit Signaling: The node that experiences congestion can explicitly send a signal to the source or destination. The explicit signaling method, however, is different from the choke packet method. In the choke packet method, a separate packet is used for this purpose; in the explicit signaling method, the signal is included in the packets that carry data.

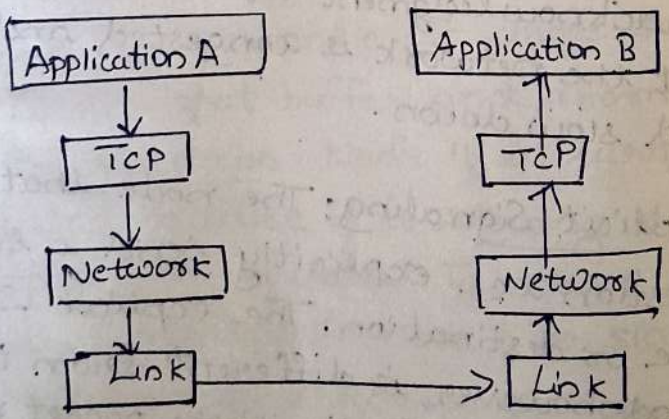
Explicit signaling can occur in either the forward or backward direction

(i) Backward Signaling: A bit can be set in a packet moving in the direction opposite to the congestion. This bit can warn the source that there is congestion

(ii) Forward Signaling: A bit can be set in a packet moving in the direction of the congestion. This bit can warn the destination that there is congestion

2.5 Flow Control

Tcp Flowcontrol is a protocol designed to manage the data flow between the user and the server. It ensures that there is a specific bandwidth for sending and receiving data so the data can be processed without facing any major issues. In order to achieve this, the Tcp protocol uses a mechanism called the sliding window protocol.



- The Sliding window protocol:

In the sliding window protocol method, when we are establishing connection

between sender and receiver, there are two buffers created. Each of these two buffers are assigned to the sender, called the sending window and to the receiver called the receiving window.

When the sender sends data to the receiver, the receiving window sends back the remaining receiving buffer space. As a result, the sender cannot send more data than the available receiving buffer space. We'll understand the concept

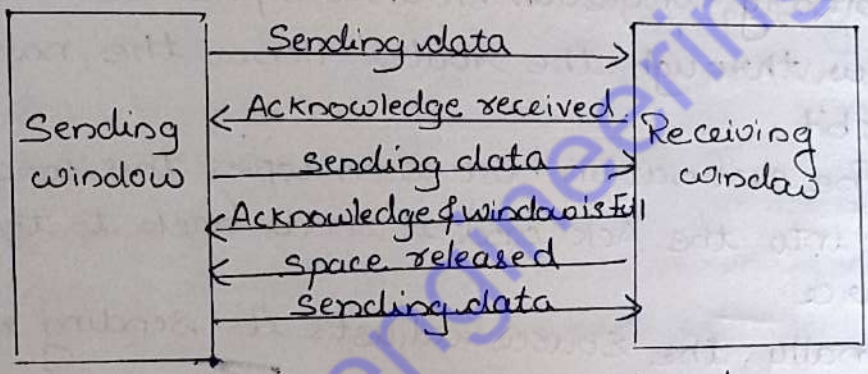


Fig: Sliding window protocol

2.6 Congestion avoidance (DECbit, RED)

To predict when congestion is about to happen and then to reduce the rate at which hosts send data just before packets start being discarded. It has three methods,

- DEC bit
- Random Early Detection (RED)
- Source based congestion control

▪ DEC Bit:

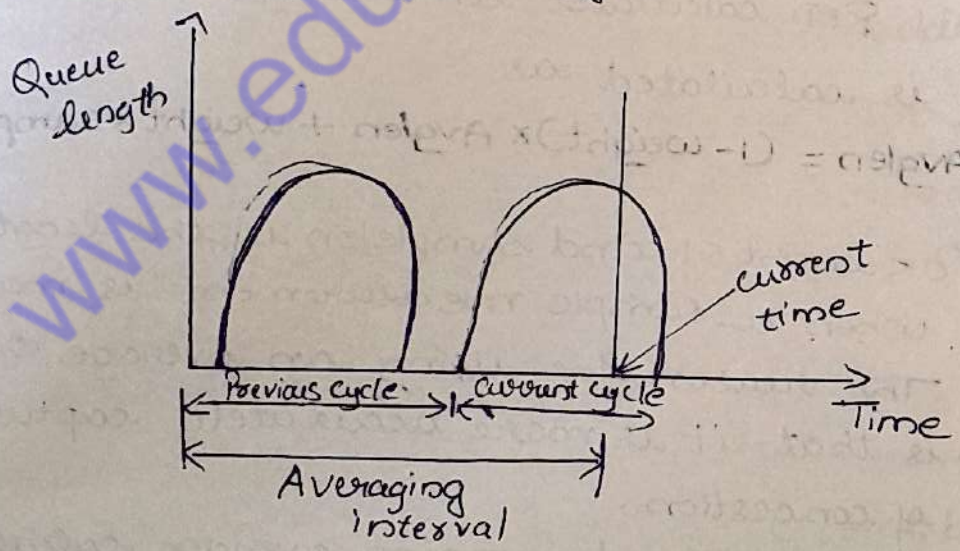
The first mechanism was developed for use on the Digital Network Architecture (DNA), a connectionless network with a connection

Oriented transport protocol. Dec bit is a Tcp Congestion control technique implemented in routers to avoid congestion. Its utility is to predict possible congestion and prevent it. When a router wants to signal congestion to the sender it adds a bit in the header of packets sent.

When a packet arrives at the router the router calculates the average queue length for the last (busy+idle) period plus the current busy period. (The router is busy when it is transmitting packets and idle otherwise).

When the average queue length exceeds 1, then the router sets the congestion indication bit in the packet header of arriving packets

This technique dynamically manages the window to avoid congestion.



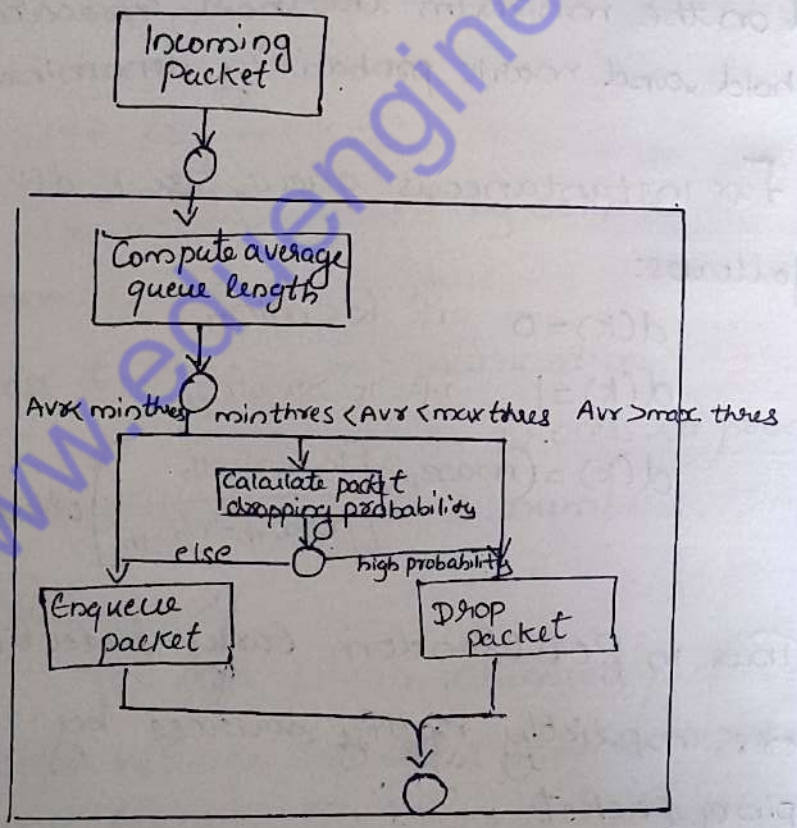
Thus In Dec bit routers explicitly notify sources about congestion.

Random Early detection

→ Random early detection (RED) also known as random early discard/drop is a queuing discipline for a network scheduler suited for congestion avoidance.

→ RED aims to control the average queue size by informing the end host to slow down the transmission of packets. It monitors the average queue size and drops packets based on statistical probabilities.

The following process is performed:



1. We have an incoming packet
2. The Average queue length is computed
3. If avr < min length threshold then the packet is placed in the queue

4. If min<avg<max queue thres then check dropping probability (Pa)

1. If high probability ⇒ packet is dropped
2. If loco probability ⇒ packet placed in the queue

5. If avg > max ⇒ Packet is dropped.

The rate of packet drop increases linearly as the average queue size increases until the average queue size reaches the maximum threshold.

The packet drop probability is based on the minimum threshold, maximum threshold and mark probability denominator.

For instantaneous queue size k, d(k) is as follows:

$$d(k) = 0 \quad \text{if } k < \min_{th}$$

$$d(k) = 1 \quad \text{if } k > \max_{th}$$

$$d(k) = (\max_p)\left[\frac{k - \min_{th}}{\max_{th} - \min_{th}}\right] \text{otherwise}$$

Thus in RED (Random Early Detection) routers implicitly notify sources by dropping packets.

2.7 SCTP (Stream Control Transmission Protocol)

→ Stream Control transmission protocol (SCTP) is a transport layer protocol, serving similar role as TCP and UDP.

→ It is a new reliable, message oriented transport layer protocol,

→ SCTP combines the best features of UDP and TCP. It preserves the message boundaries, and at the same time, detects lost data, duplicate data and out of order data.

→ It has congestion and flow control mechanism.

2.7.1 SCTP Services:

The services offered by SCTP to the application layer processes are as follows:

1. Process to process communication:

SCTP provides process to process communication using port numbers.

2. Multistreams:

TCP is a stream-oriented protocol. Each connection between TCP client and a TCP server involves one single stream. The problem with this approach is that a loss at any point in the stream blocks the delivery of rest of the data.

SCTP allows multi stream service in each connection, which is called association in SCTP terminology.

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

3. Multihoming:

A TCP connection involves one source and one destination IP address. This means that even if the sender or receiver is a multihomed host (connected to more than one physical address with multiple IP addresses), only one of these IP addresses per end can be utilized during the connection.

But SCTP supports multihoming service. The sending and receiving host can define multiple IP addresses in each end for an association.

In this fault tolerant approach, when one path fails another interface can be used for data delivery without interruption.

This feature is very helpful when we are sending and receiving a real time payload such as internet telephony.

4. Full duplex communication:

Like TCP, SCTP offers full duplex services in which data can flow in both directions at the same time.

Each SCTP has a sending and receiving buffer and packets are sent in both directions

5. Connection Oriented Service

In SCTP, a connection is called an association. When a process at site A wants to send and receive data from another processes at site B, the following occurs:

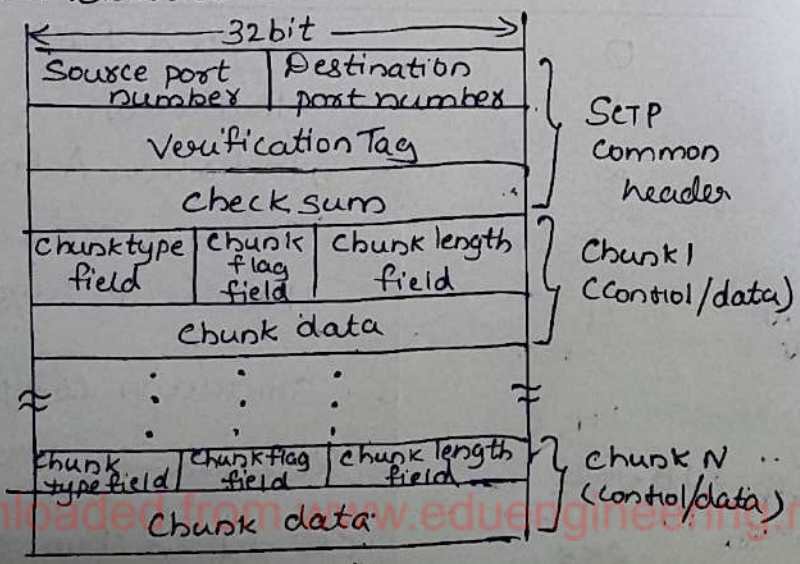
1. The two SCTPs establish an association between each other.
2. Data are exchanged in both directions.
3. The association is terminated.

6. Reliable Service :

Like TCP SCTP is a reliable transport protocol. It uses an acknowledgement mechanism to check the safe and sound arrival of data.

2.7.2 SCTP Packet format :

SCTP transmit data in the form of messages and each message contains one or more packets. The control chunks come before data chunks.



- General header

A SCTP packet contains a common header and one or more chunks. The SCTP common header contains the following information.

1. Source and destination port numbers to enable multiplexing of different associations at the same address.
2. A 32 bit verification tag that guards against the insertion of an out-of-date or false message into the SCTP association.
3. A 32 bit checksum for error detection

- Chunk Layout

1. A chunk can be either a control chunk or a data chunk. A control chunk incorporates different flags and parameters. Data chunk incorporates flags to control segmentation and reassembly
2. Chunk type field identifies the type of information contained in the chunk data field. SCTP consists of one Data chunk and 12 control chunks.

Chunk number	Chunk name
0	Payload Data
1	Initiation
2	Initiation Acknowledgment
3	Selective acknowledgement
4	Heartbeat request
⋮	⋮
14	Shutdown complete
15-62	Reserved for IETF
⋮	⋮
255	IETF-defined chunk extensions

3. Chunk length field represents the size of the fields chunk type, chunk flag, chunk length and chunk value in bytes.

4. Chunk data are used to send actual data through the stream.

2.8 Quality of Services

Quality of service (Qos) is basically the ability to provide different priority to different applications in order to guarantee a certain level of performance to the flow of data.

Qos is basically the overall performance of the computer network.

Given below are four types of characteristics that are mainly attributed to the flow and these are as follows:

- Reliability
- Delay
- Jitter
- Bandwidth

→ Reliability:

It is one of the main characteristics that the flow needs. If there is a lack of reliability then it simply means losing any packet or losing an acknowledgement due to which retransmission is needed. Reliability becomes more important for electronic mail, file transfer & for internet access

→ Delay:

Another characteristic of the flow is the delay in transmission between the source and destination. During audio conferencing, telephony, video conferencing there should be a minimum delay.

→ Jitter:

It is basically the variation in the delay for packets that belongs to the same flow. Thus jitter is basically the variation in the packet delay. Higher the value of jitter means there is a large delay and the low jitter means the variation is small.

→ Bandwidth:

The different application need different bandwidth

Types of Quality of Service Solutions:

1. Stateless Solution: Here, the server is not required to keep or store the server information or session details to itself. The routers maintain no fine grained state about traffic. also it has weak services as there is no guarantee about the kind of performance delay. In the stateless solution, the server and client are loosely coupled

2. Stateful Solution: Here, the server is required to maintain the current state and session information, the routers maintain per flow state as the flow is very important in providing the Quality of service which is providing powerful services such as guaranteed services. Here the server and client are tightly bounded.

- Quality of service parameters

Qos can be measured quantitatively by using several parameters:

→ Packet loss
→ Jitter
→ Latency
→ Bandwidth

- Techniques to improve Qos

Generally, there are four techniques to improve quality of service.

→ Scheduling
→ Traffic Shaping
→ Resource Reservation
→ Admission control

- Scheduling:

Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner.

Several Scheduling techniques are designed to improve the quality of service. They are

- FIFO Queuing
- Priority queuing
- Weighted fair queuing

- **Traffic Shaping**

It is a mechanism to control the amount and the rate of the traffic sent to the network. Two techniques can shape traffic: leaky bucket
token bucket

- **Resource Reservation**

A flow of data needs resources such as a buffer, bandwidth, cpu time and so on. The quality of service is improved if these resources are reserved beforehand.

- **Admission control**

Admission control refers to the mechanism used by a router or a switch to accept or reject a flow based on predefined parameters called flow specifications. Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity can handle new flow.

Question bank

1. What are the advantages of using UDP over TCP (Dec16)
2. Give the approaches to improve the QoS. (May11, Dec17)

   1. Fine grained approaches, which provide QoS to individual applications or flows.

   2. Coarse-grained approaches, which provide QoS to large classes of data traffic.

3. What is TCP? (Dec11)
4. Define congestion. (Dec 11)
5. What do you mean by slow start in TCP congestion? (May 16)

   Slow start is part of the congestion control strategy used by TCP. Slowstart is used in conjunction with other algorithms to avoid sending more data than the network is capable of transmitting.

6. What do you mean by QoS? (Dec 14, 15, 16, 18)
7. Suppose TCP operates over a 1-Gbps link, utilizing the full bandwidth continuously. How long will it take for sequence numbers to wrap around completely? Suppose an added 32 bit timestamp field increments, 1000 times during this wrap around time, how long will it take for the timestamp field to wrap around? (May 13, 18)

   TCP Advertised Window is 16 bits,
   Sequence number is 32 bit

   So there will be 2^{32} bytes on the fly in this 1 Gbps link.

The corresponding transmission time is
$2^{32} \times 8 / 1 \times 10^9 = 34.36$ sec
So it will take 34.36 sec to wrap around the sequence number
Each increment of time stamp = 34.36 sec / 1000
= 34.36 ms
So the total time can be expressed by this
timestamp = $34.36 \times 10^{-3} \times 2^{32}$ sec
$= 1.48 \times 10^8$ sec = 4.68 year
So by adding this timestamp, it will take 4.68 year to wrap around the sequence number.

8. Differentiate between delay and jitter? (Dec
9. List some ways to deal with congestion?
10. Differentiate UDP and TCP (May 14, 16)
11. What are the services provided by transport layer protocol
12. Define congestion control

EDU ENGINEERING

PIONEER OF ENGINEERING NOTES

TAMIL NADU'S BEST EDTECH PLATFORM FOR ENGINEERING

CONNECT WITH US

WEBSITE: www.eduengineering.net

TELEGRAM: @eduengineering

INSTAGRAM: @eduengineering

- **Regular Updates for all Semesters**
- **All Department** Notes AVAILABLE
- **Handwritten** Notes AVAILABLE
- **Past Year** Question Papers AVAILABLE
- **Subject wise** Question Banks AVAILABLE
- **Important Questions** for Semesters AVAILABLE
- Various **Author Books** AVAILABLE