UNIT-VII
TRANSPORT LAYER

↳ Process to process delivery
↳ UDP & TCP protocols
↳ SCTP
↳ Data Traffic
↳ Congestion
↳ Congestion control
↳ QoS
↳ Integrated services
↳ differentiated service
↳ QoS in Switched Networks

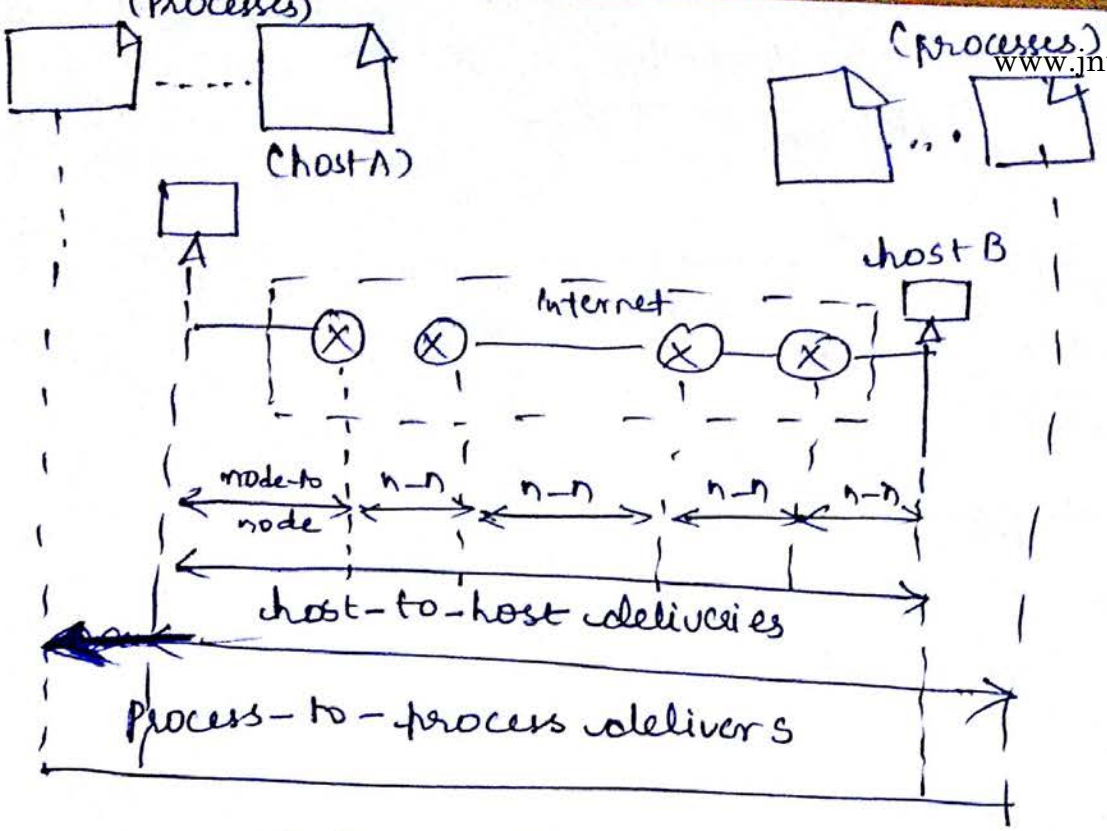
* Services provided by Transport Layer are:

1) connection-Oriented communication
2) Reliability
3) flow control
4) Congestion avoidance
5) Multiplexing.

Process-to-process delivery:

Transport Layer is responsible for process-to-process delivery. Real communication takes place b/w two processes (application programs).

↳ The processes communicate in a client/server relationship
-P.
↳ The following figure explain about the types of data deliveries.



a) figure: Type of data deliveries

↳ To acheive process-to-process communication, the Client/server paradigm is used i.e A process on the local host is called client, & process on the Remote host is called Server. So, for communication to happen, we need

- ↳ local host
- ↳ local process
- ↳ Remote host
- ↳ Remote process.

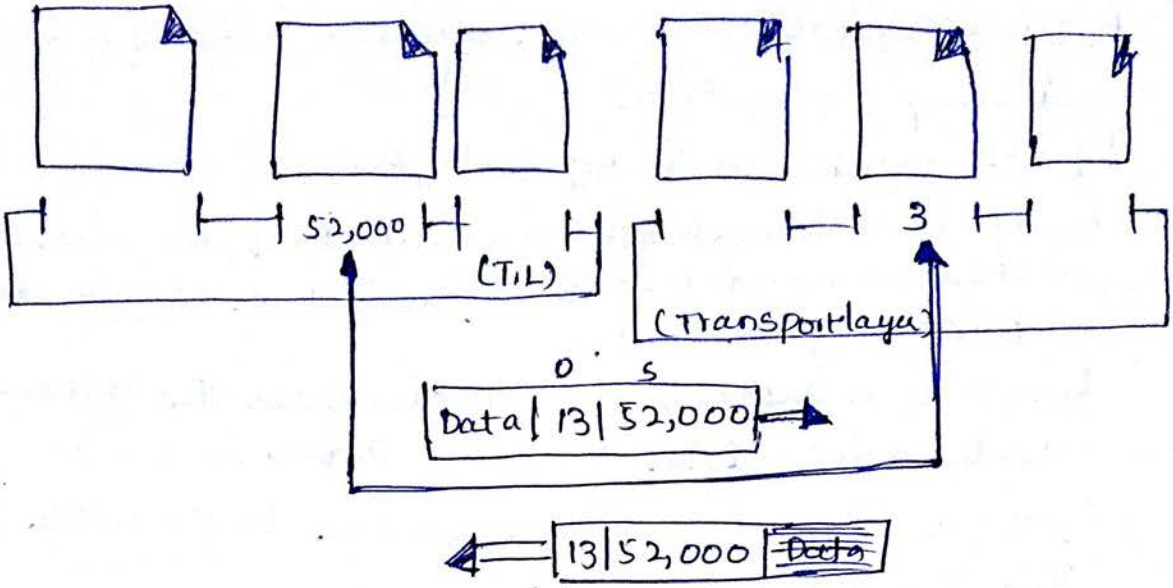
Addressing:

↳ Transport layer uses "Port-addres" to have process-to-process communication.

↳ Both source & destination processes should have the port numbers. The destination port number is needed for delivery & the source port number is needed for reply.

↳ In OSI model, the range of port numbers is from 0 - 65,535 which uses 16-bit Integers to represent.

Ex: (Daytime client) (Daytime Server)



↳ The IANA (Internet Assigned Number Authority) has divided the port numbers into three ranges:

① Well-known ports: (0-1023, which are controlled by IANA)

② Registered ports: (1024-49,151, used to prevent duplication) & are not assigned & controlled by IANA.

③ Dynamic ports: (49,152-65,535, they are neither controlled/registered. they can be used by any process)
(or) ephemeral ports

Socket Addresses:

process-to-delivery needs two identifiers, IP address and the port number, at each end to make a connection Therefore the combination of an IP-address & a port NO is called a Socket address

Socket address = IP address + Port No.

Ex:

IP Address [200.23.56.8] [69] Port Number

Socket Address [200.23.56.8 69]

TCP (Transmission Control protocol)

↳ Tcp is reliable, connection-oriented, Ordered & error-checked delivery of a station.

↳ Tcp resides in Transport layer

↳ Tcp is a Connection-Oriented protocol, because it first establi-shes an end-to-end communication session before any data may be send.

↳ TCP is a protocol that make sure the data has been well delivered, in the correct order.

↳ There is also sequence number to assemble the packets in the original order.

↳ Packets may use different paths to reach the recipient, or a corrupted Packets need to be resend.

↳ i.e the Recipient might get the Pack in the corong order, the sequence number make sure when reassembling, Packets are in the correct order.

↳ One other interesting feature of TCP is the windows handling. The rate of data transmission b/w two devices is managed by a windowing system to prevent a fast sender from transmitting more data than can be supported by the receiving.

↳ TCP is a connection set-up,

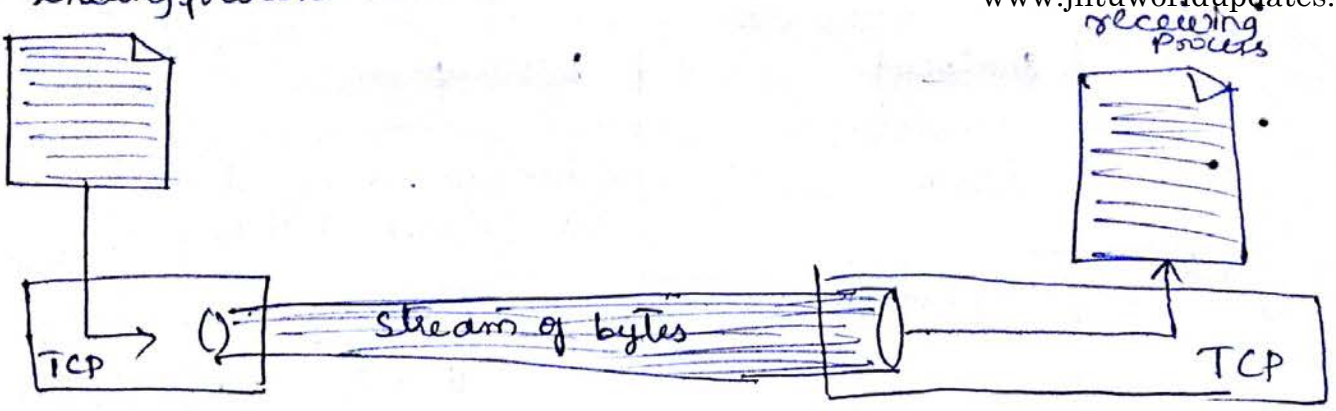
Coliscarding of corrupted packets,
Retransmission of lost packets,
flow control,
Congestion control.

↳ TCP provides process-to-process Communication using Port numbers.

↳ The following table represents the list of some of well-known Port numbers used by TCP.

port	protocol	Description
7	Echo	echoes an received datagram back to the sender
9	discard	discards any datagram that is received.
11	Users	Active users
13	Daytime	Returns the data & time
17	quote	Return the quote of the day
19	chargen	Returns a string of characters.
20	FTP, data	File Transfer protocol (data connection)
21	FTP, control	File Transfer protocol (control connection)
23	TELNET	Terminal N/w
25	SMTP	Simple Mail Transfer protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap protocol
79	FINGER	Finger
80	HTTP	Hypertext Transfer protocol
111	RPC	Remote procedure call

↳ Stream delivery is done by TCP i.e.,



↳ TCP is a connection-oriented service, i.e.,

(1) The two TCP's establishes a connection b/w them
(2) Data are exchanged in both directions
(3) The connection is terminated.

↳ TCP offers full-duplex service in which data can flow in both directions at the same time.

↳ Each TCP has a sending buffer & receiving buffers & segments move in both directions.

↳ TCP is a reliable transport protocol, It uses an acknowledgment mechanism to check the safe & sound arrival of data.

↳ TCP uses Numbering systems (sequence Number & acknowledgement number), to keep track of the segments being transmitted (or) received in an order.

↳ TCP provides flow control, the receiver of the data controls the amount of data that are to be sent by the sender. This is done to prevent the receiver from being overwhelmed with data. The numbering system allows TCP to use a byte-oriented flow control.

↳ Error control is done by TCP, to provide reliable services.

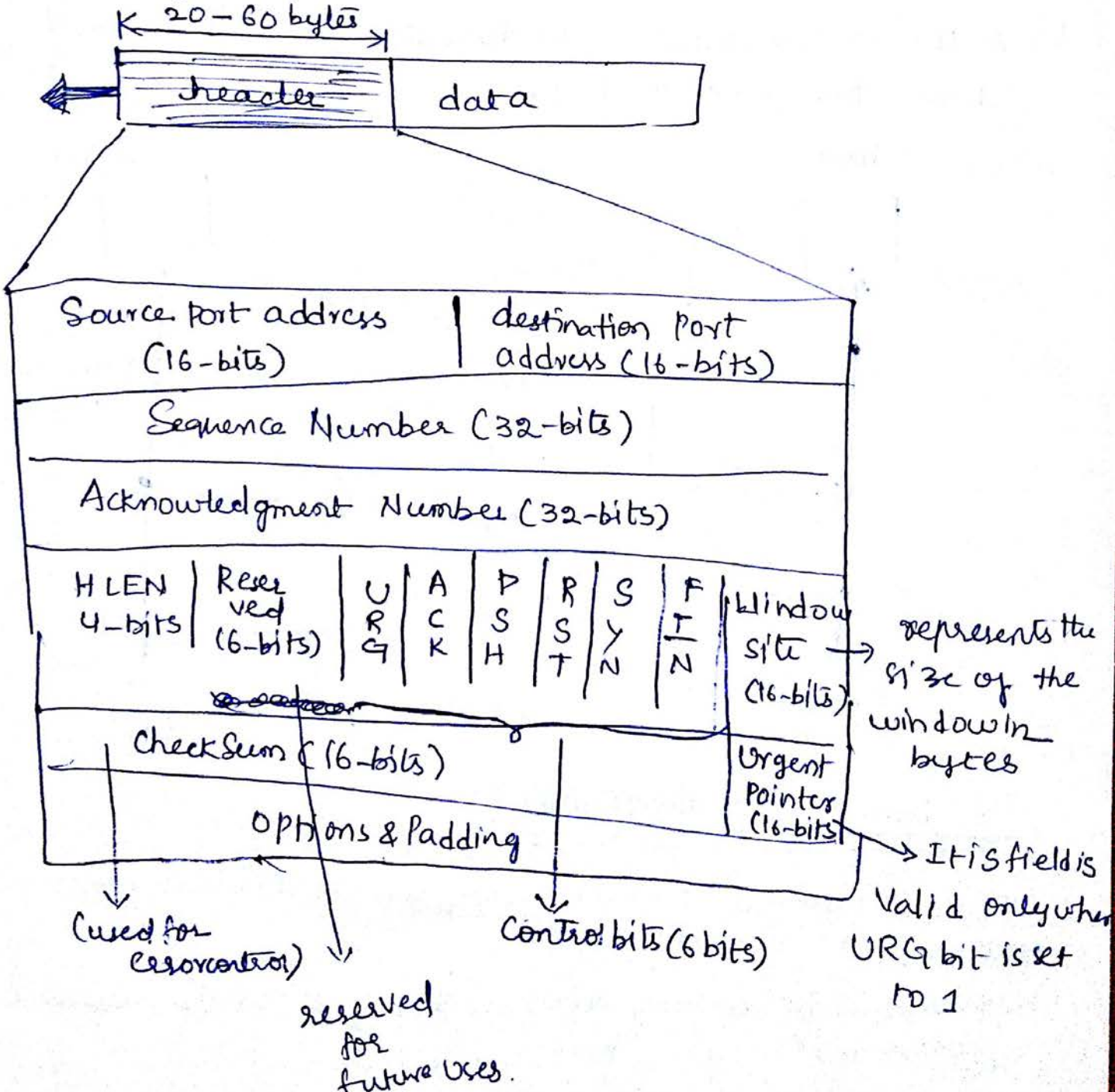
↳ TCP does congestion control in the N/w, i.e., the amount of data sent by a sender is not only controlled by the

receives (flow control), but is also determined by the level of congestion in the N/w.

↳ the data in the transport layer is known as "segments".

↳ The TCP segment format is shown below, which has two fields [header + data]

* TCP segment format:



URG: Urgent point is Valid

ACK: acknowledgment is Valid

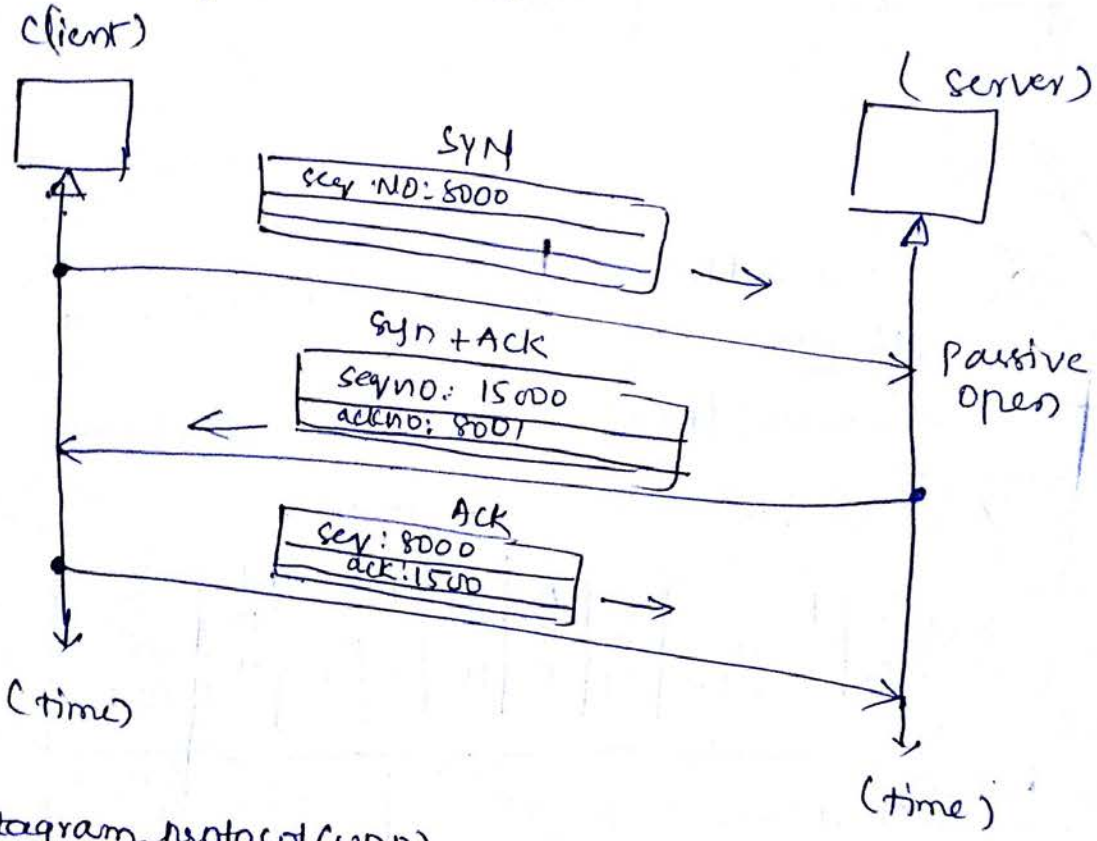
PSH: Request for push (push the data)

RST: Reset the connection.

SYN: Synchronize sequence Numbers (during connection)

FIN: Terminates the connection.

↳ A TCP connection establishment, uses three-way handshaking method. i.e.,



User Datagram protocol (UDP)

UDP is called a connectionless, Unreliable transport protocol.

↳ UDP doesnot do any services, simply it does process-to-process delivery & does limited error checking

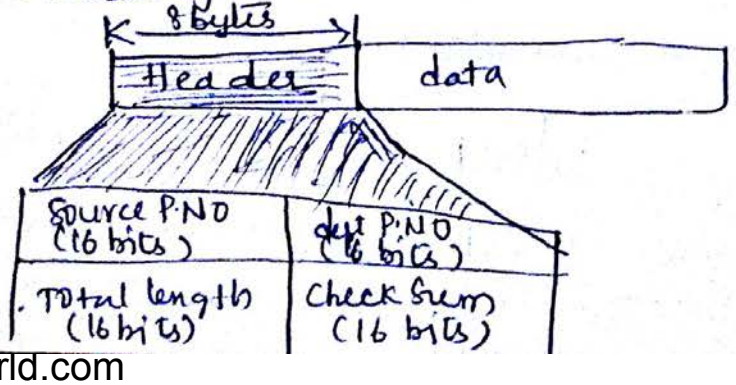
↳ UDP is very simple protocol using a minimum of overhead.

↳ If a process want to send a small message & doesnot care much about reliability, we can then use UDP.

↳ Well-known ports of UDP are:

Port	Protocol	Description
7	Echo	Echoes a received datagram back to sender
9	Discard	Discards any datagram that is received
11	Users	Active Users
13	Daytime	Returns the data & the time.
17	quote	Returns the quote of the day.
19	chargen	Returns a string of characters.
53	Nameserver	Domain Name Service
67	BOOTPS	Server port to download bootstrap Information
68	BOOTPC	client port to download bootstr-ap Information.
69	TFTP	Trivial file Transfer protocol.
111	RPC	Remote procedure call
123	NTP	Network time protocol
161	SNMP	Simple N/w management protocol
162	SNMP	Simple N/w management protocol (trap)

↳ User datagram format is shown as:



⇒ UDP Operation's are:

① It provides a connectionless services, here each user datagram sent by UDP is an independent datagram (i.e., the user datagrams are not numbered), and also there is no connection establishment and no connection termination.

② There is no flow control, so that the receiver may overflow with incoming messages.

③ there is no error control, except checksum. when the receiver detects an error through the checksum, the user datagram is silently discarded.

↳ Uses of the UDP protocol are:

① UDP is suitable for a process that requires simple request response communication, and with little flow & error control.

② UDP is suitable for process with internal flow & error control mechanisms.

③ UDP is suitable for multicasting

④ UDP is used for management process, such as SNMP.

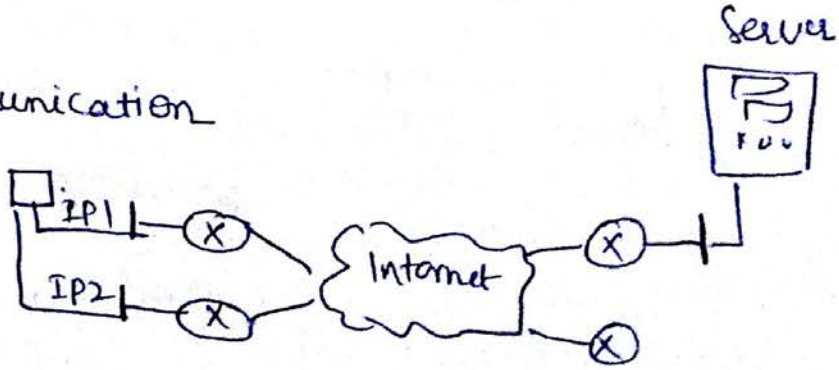
⑤ UDP is used for some route updating protocols, such as RIP (Routing Information protocol).

SCTP (Stream Control Transmission protocol):

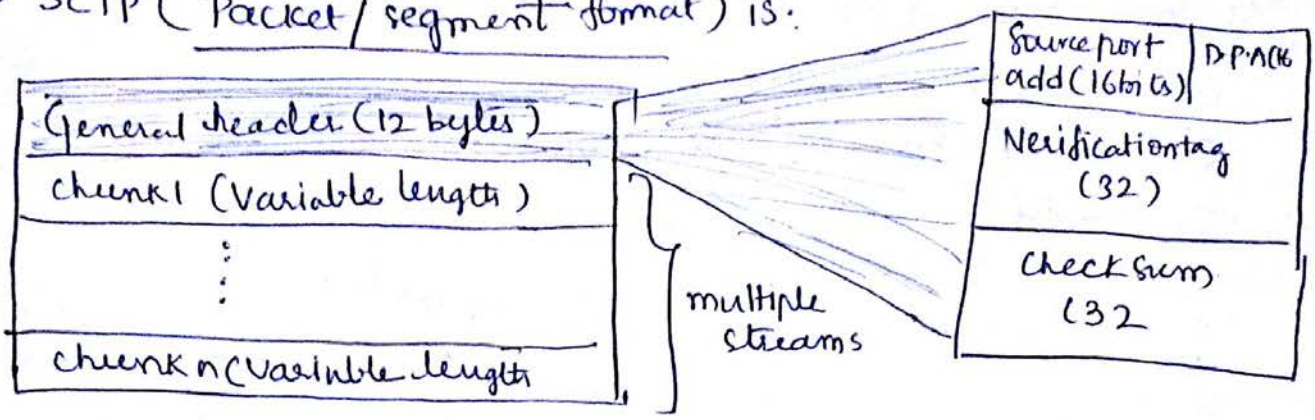
SCTP is a message oriented, reliable protocol that combines the best features of UDP & TCP.

↳ SCTP services are:

① process-to-process communication
② multiple-streams
③ multi-homing ⇒
④ full-duplex communication
⑤ Connection-oriented service
⑥ Reliable service



⇒ SCTP (Packet / segment format) is:



⇒ SCTP, like TCP, is a reliable Transport Layer protocol. It uses a SACK chunk to report the state of the receiver buffer to the sender, & performs error control.

↳ SCTP, like TCP performs flow control.

↳ SCTP, like TCP performs congestion control in the N/w SCTP has slow start (exponential increase), congestion avoidance (additive increase), and congestion detection phases.

like TCP, SCTP also uses fast retransmission & fast recovery.

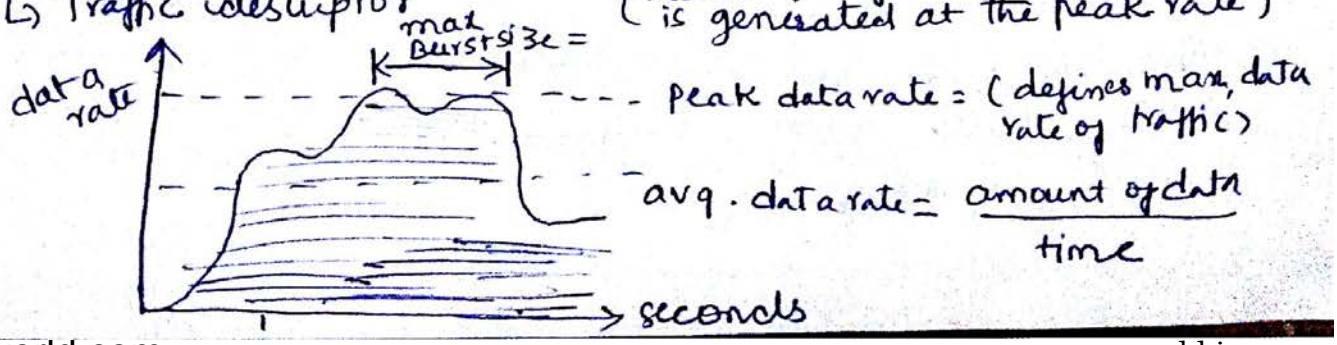
Data Traffic

↳ The Main focus of congestion control & quality of service (QOS) is data traffic.

↳ In congestion control, we try to avoid traffic congestion.

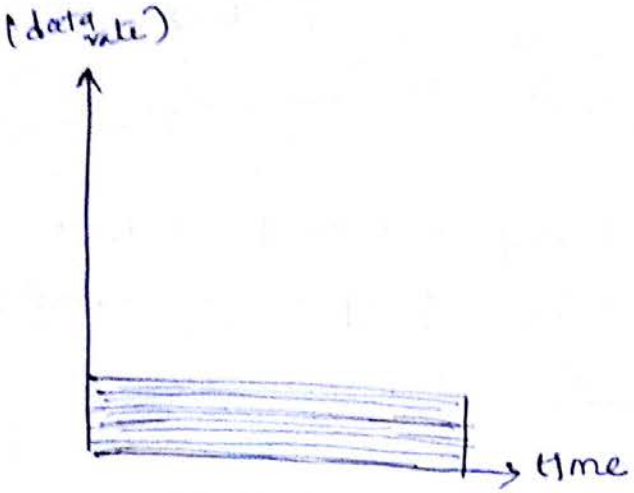
↳ In QOS, we try to create an appropriate environment for the traffic.

↳ Traffic descriptor

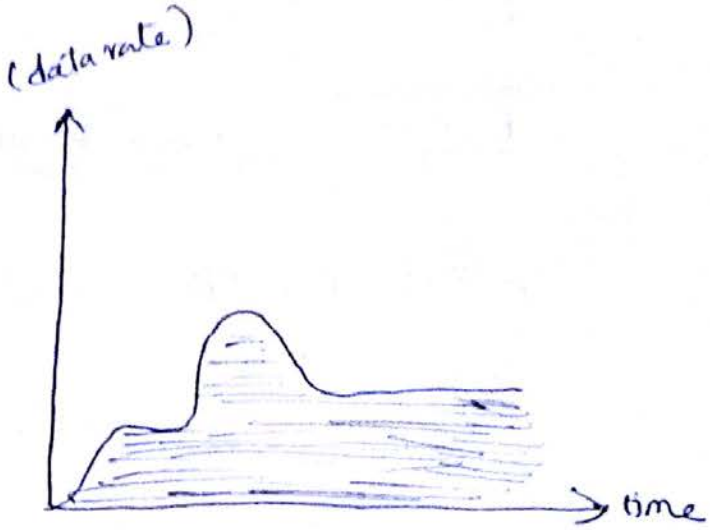


⇒ Three traffic profiles:

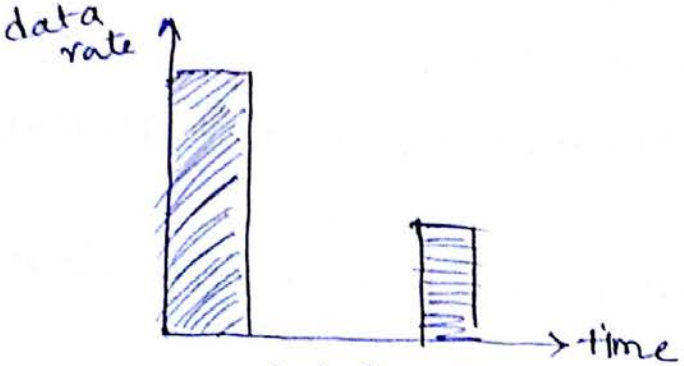
1. Constant Bit rate (CBR)
2. Variable bit rate, and (VBR)
3. Bursty



(a) CBR



(b) VBR



(c) Bursty

Congestion & congestion control:

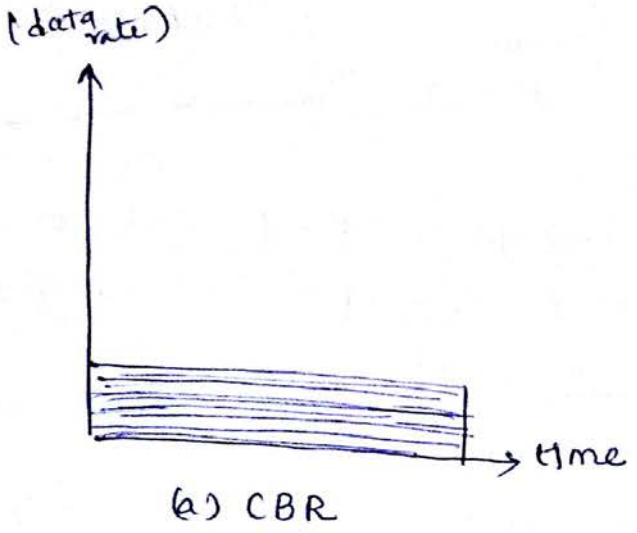
Congestion: congestion in the N/w may occur if the load on the N/w, the no. of packets sent to the N/w is greater than the capacity of the N/w.

↳ congestion control refers to mechanisms & techniques to control the congestion & keep the load below the capacity.

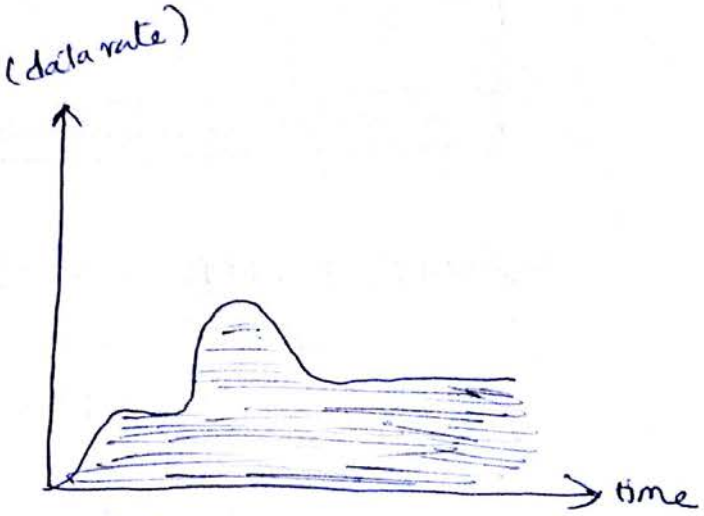
↳ congestion in a N/w/ Internetwork occurs because routers & switches have queues - buffers that hold the packets before & after processing.

⇒ three traffic profiles:

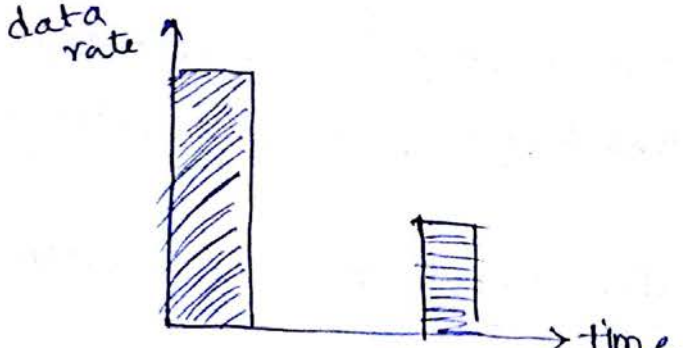
① Constant Bit rate (CBR)
② Variable bit rate, and (VBR)
③ Bursty



(a) CBR



(b) VBR



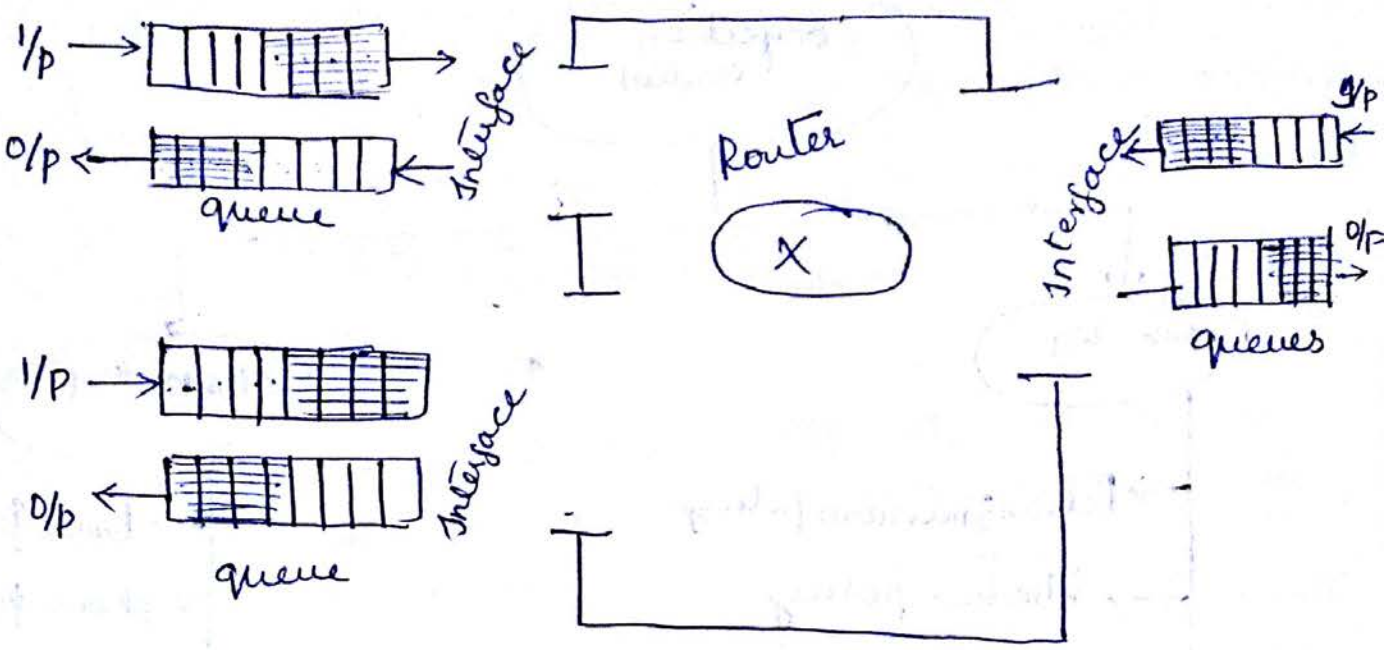
(c) Bursty

Congestion & congestion control:

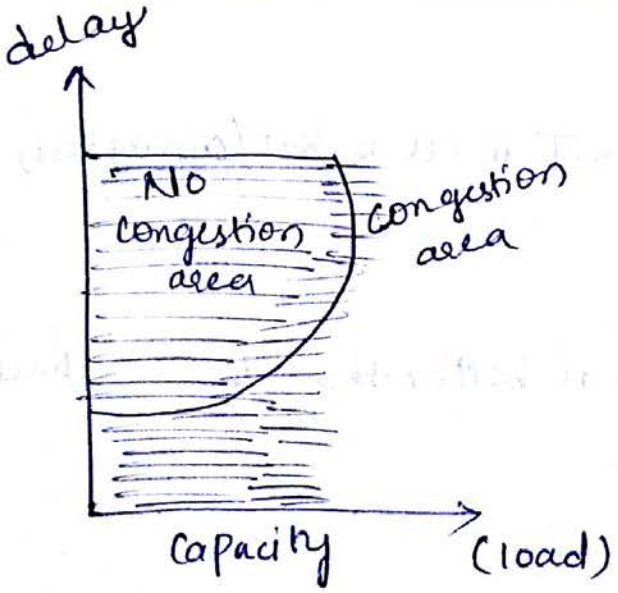
Congestion: congestion in the N/w may occur if the load on the N/w, the no. of packets sent to the N/w is greater than the capacity of the N/w.

↳ congestion control refers to mechanisms & techniques to control the congestion & keep the load below the capacity.

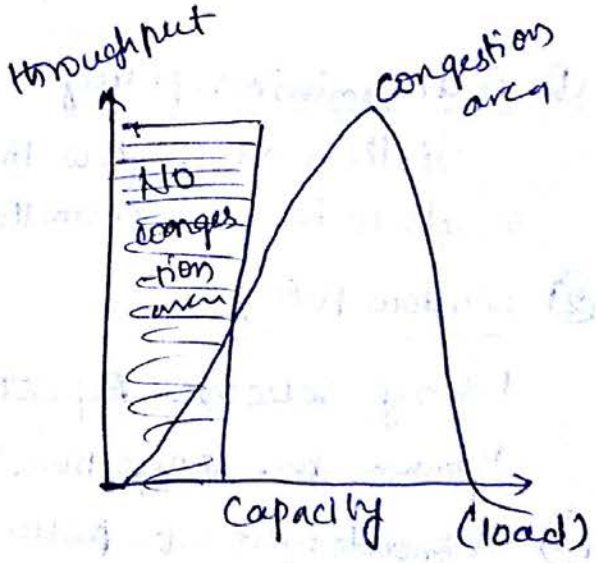
↳ congestion in a N/w internetwork occurs because routers & switches have queues - buffers that hold the packets before & after processing.



↳ Congestion control, involves two factors that measure the performance of a N/w: delay & throughput



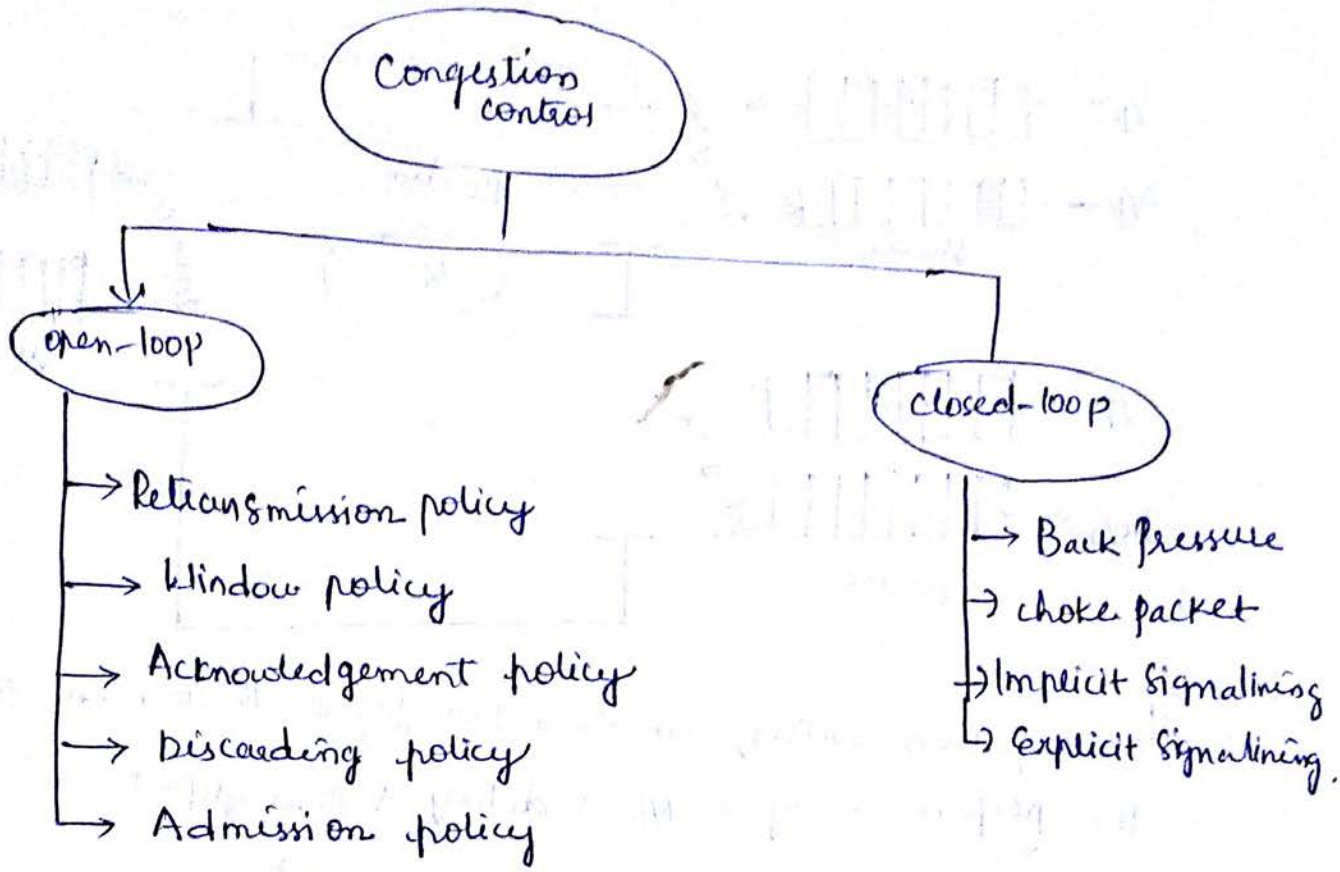
(a) delay as a function of load.



(b) Throughput as a function of load.

Congestion control:

It refers to techniques & mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened.



① Retransmission policy:

If the sender feels that the sent packet is lost/corrupted, the packet needs to be retransmitted.

② Window policy:

Using Selective-Repeat window is better than the Go-Back-N window for congestion control.

③ Acknowledgement policy:

If the receiver doesnot acknowledge every packet it receives, it may slow down the sender & help prevent congestion.

④ Discarding policy:

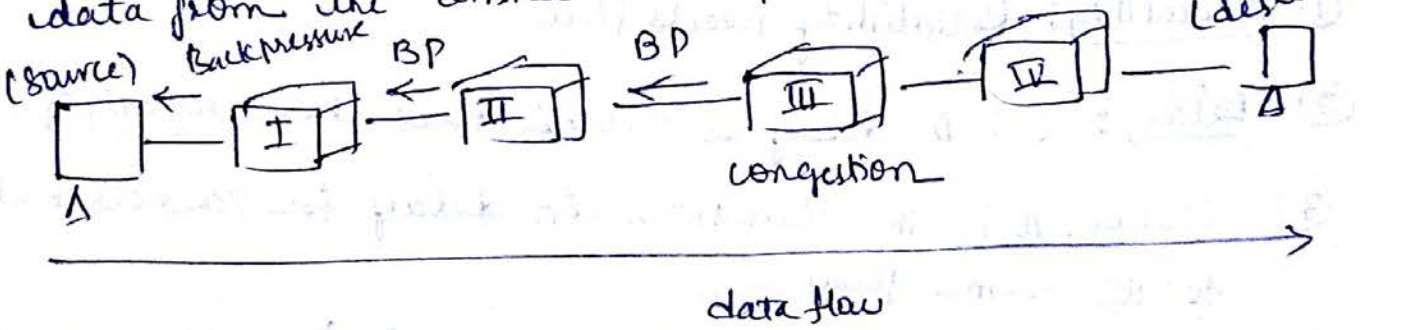
A good discarding policy by the routers may prevent congest-ion & at the same time, may not harm the integrity of the transmission.

⑤ Admission policy:

It is a qos mechanism, can also prevent congestion in VCN's. Switches in a flow first check the resource requirement of a flow before admitting it to the N/w.

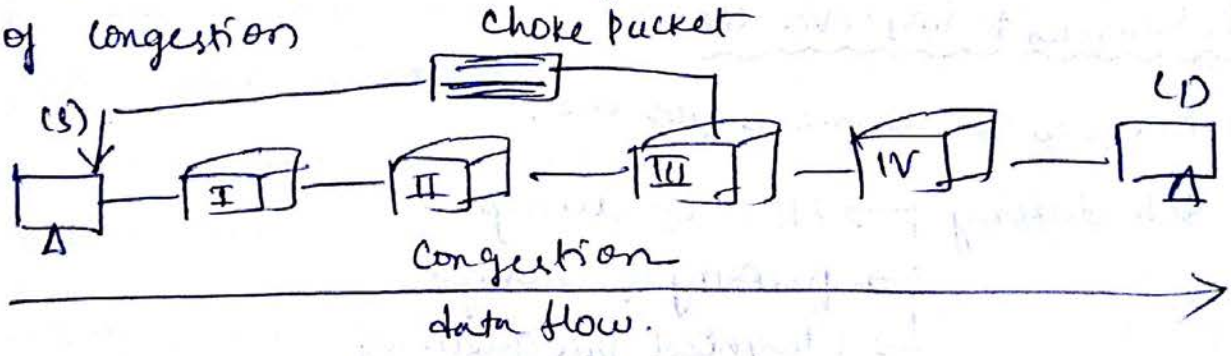
⑥ Back pressure:

This technique of back pressure refers to a congestion control mechanism in which a congested node stops receiving data from the immediate upstream nodes / nodes.



⑦ Choke packet:

It is a packet sent by a node to the source to inform it of congestion



⑧ Implicit Signaling:

Here, there is no communication b/w the congested node/ nodes & the source. The source guesses that there is a congestion somewhere in the N/w from other symptoms.

⑨ Explicit Signaling:

Here, the node that experiences congestion can explicitly send a signal to the source/destination.

Quality of Service (QOS):

The flow characteristics of QOS are:

1. Reliability
2. Delay
3. Jitter
4. Bandwidth

1. **Reliability:** Reliability needs flow
2. **Delay:** (S-D delay is another flow characteristic)
3. **Jitter:** It is the variation in delay for packets belonging to the same flow.
4. **Bandwidth:** Different applications need different Bandwidths (Video/audio) so effects on flow.

Techniques to improve QOS:

Techniques to improve QOS are:

- Scheduling
  - FIFO Queuing
  - Priority Queuing
  - Weighted fair queuing
- Traffic shapping
  - Leaky Bucket
  - TOKEN Bucket
- Admission control
- Resource Reservation

Integrated Services:

↳ Integrated Services is a flow-based QoS model designed for IP.

↳ There are two models designed to provide QoS in the internet. They are:

- ↳ Integrated Services
- ↳ Differentiated Services

↳ In Integrated Services, the user needs to create a flow, a kind of virtual circuit, from the source to the destination & inform all routers of the resource requirement.

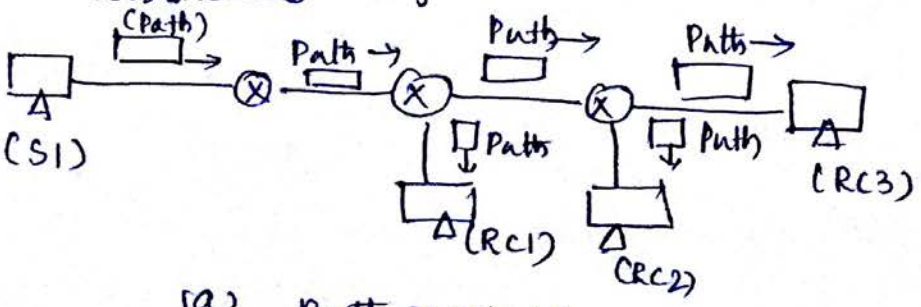
↳ IP is a connection-less datagram, Packet-Switching protocol when flow-based model can't be implemented, the solution is to use a signaling protocol to run over IP that provides the signaling mechanism for making a reservation. This protocol is called RSVP (Resource Reservation protocol).

↳ In Integrated service model, an application program needs resource Reservation (to control flow).

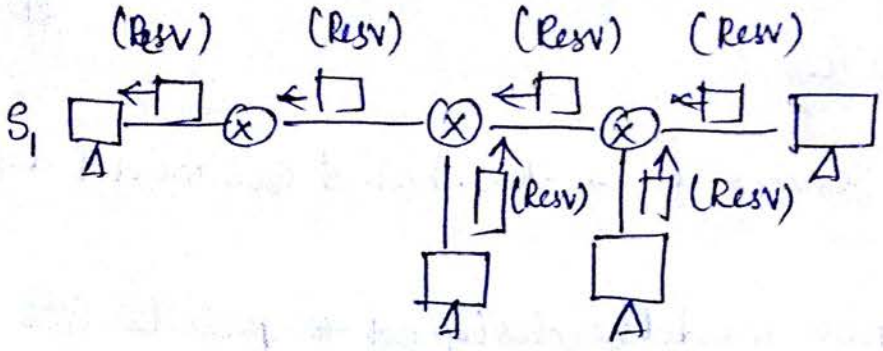
↳ RSVP is a signalling protocol to help IP create a flow & consequently make a resource reservation.

↳ RSVP messages:

- ↳ Path messages
- Reservation messages.



(a) Path message



b) Resv messages.

↳ Reservation styles:

- ↳ Wild card filter (WF)
- ↳ fixed filter (FF)
- ↳ Shared Explicit (SE)

↳ problems with Integrated Services:

- ↳ Scalability
- ↳ Service-type limitation.

Differentiated Services:

↳ Differentiated Service is a class-based QoS model designed for IP.

↳ DS field is shown as:-

→ currently unused.

↓
differentiated Services code Point

QOS in switched N/w

Frame Relay & ATM, these two N/w are Virtual circuit N/w's that need a signaling protocol such as RSVP.

⇒ QOS in frame Relay

- Access time
- Committed Burst Size (BC)
- committed information Rate (CIR)
- Excess Burst Size (Be)

⇒ QOS in ATM

↓

Service classes

- CBR (Constant Bit Rate)
- NBR (Variable Bit Rate)
- ABR (Available Bit Rate)
- UBR (Unspecified Bit rate).