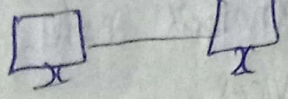
4. Transport layer

* In OSI - 7 layer 4th layer is transport layer.
* This Transport layer provides a communication services x blw a computers connected in a network.

ex:



providing communication service blw computer connected in network

* The main aim of transport layer is Process to process delivery

* **Process to process delivery**

* if you take Data link layer it deliver frames blw neighbouring nodes over a link
* DL layer helps to delivery frames blw nodes over a link.
* so it is called node to node delivery → in DL Layer

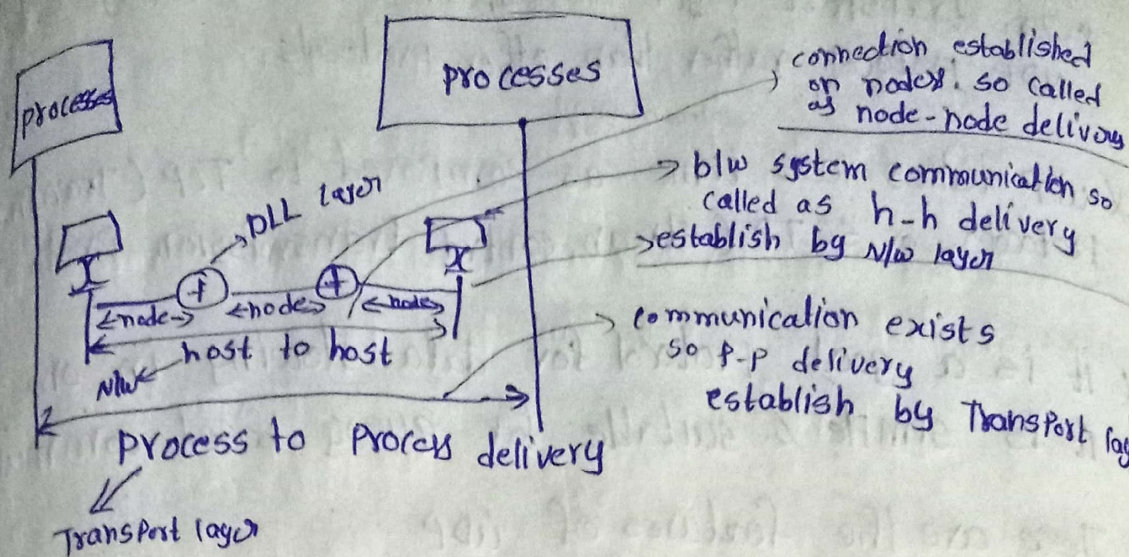
2 Network layer

* The Network layer helps to deliver a datagram blw hosts
* we call it is a host to host delivery

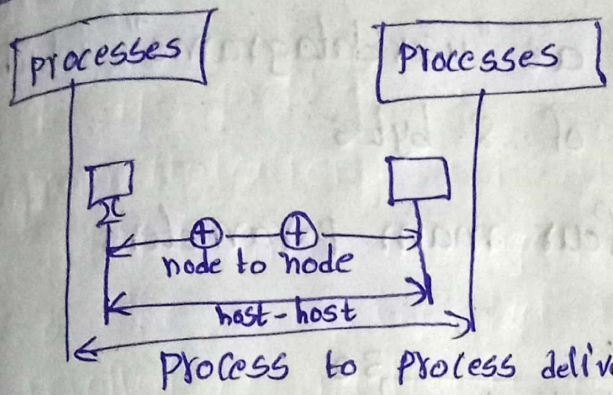
Transport

3) ~~Network~~ layer

* Transport layer helps to delivery packet or part of msg from 1 process to another process
* so we called as process to process delivery.



next drawn



UDP

* let see the protocols of transport layer i.e UDP TCP.
* The first protocol is user datagram protocol (UD
* UDP protocol is called a connectionless, unreli protocol.
* it has very limited error checking capability fr why it is called unreliable
* It is a very ~~single~~ simple protocol and it can be with minimum overhead.
* UDP can be used, when process needs to send o small msg without any issue of reliability

* In transport layer transport small message it uses UDP protocol rather than other protocols.
→ because small msgs.

* UDP takes less time as compared to TCP (Transmission Protocol) or SCTP (Stream contrd transmission Protocol)

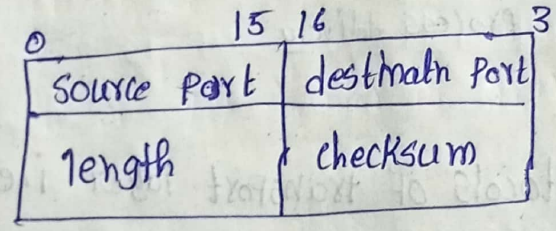
* it is a good protocol for data flowing in one direction
* it is simple & suitable for query based communicatio

* These are the features of UDP

User Datagram Header

* UDP packets are called as "user datagrams" which co the fixed-size header of 8-bytes

* UDP header contains four main parameters



→ 32 bit header divide into 4 main parts

Source Port :- This is 16-bit information is used to identify the source port of the packet.

Destination Port :- This 16-bit information, is use to identify application level services on destination machine.

Length :- length field specifies the entire length of UDP packet (including header). It is a 16-bit field-bit field & minimum value is 8-byte, ie the

size of UDP header itself.

checksum :- This field stores the checksum value generated by the sender befor sending.

* IPv4 has this field optional. so when checksum field does not contain any value it is made 0 & all its bits are set to zero.

* This is the header format of the udp protocol.

TCP (transmission control protocol)

* TCP is connection oriented protocol
* it uses flow & error control mechanism at transport layer, Hence it is reliable transport Protocol.
* TCP ensures that the data reaches intended destination in the same order it was sent.
* the sending order is same in recieved order.
* In TCP, a segment carries a data and control information.

Connection establishment :-

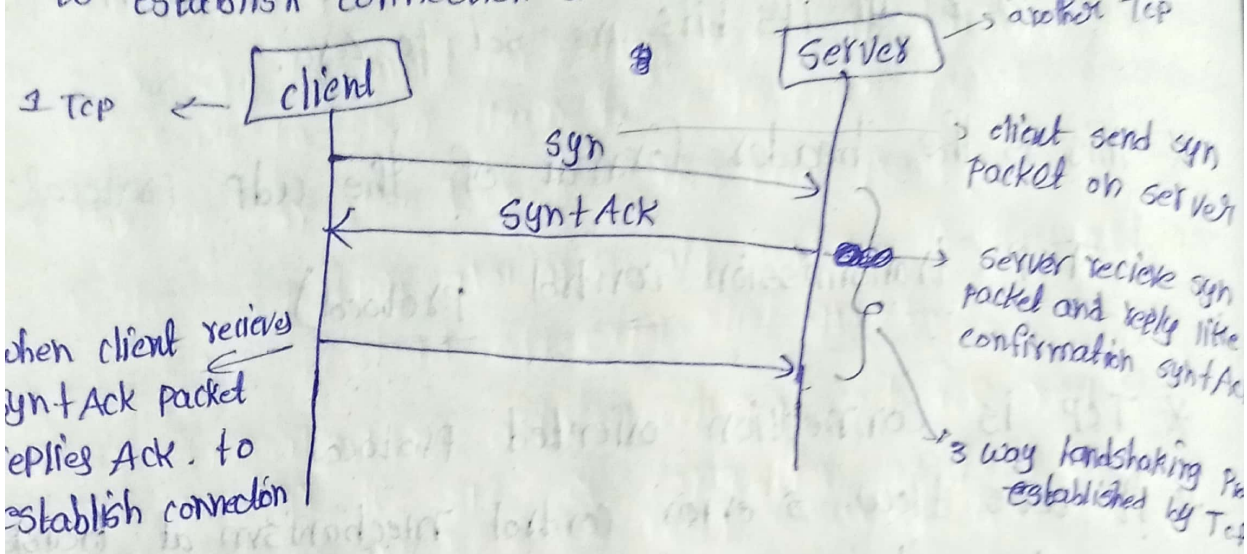
* TCP transmits data in both direction (full duplex mode) ⟷
* when two TCP's are connected to each other, each TCP to intialize the communication (SYN) and Acknow-ledgment (SYN → synchronization)
* whenever transmision control of 2 protocols has to initialize the communication with synchronization (SYN)

and approval (Ack) from each end to send the data.

ex: first i say Hello you say Hello, what can i do like that

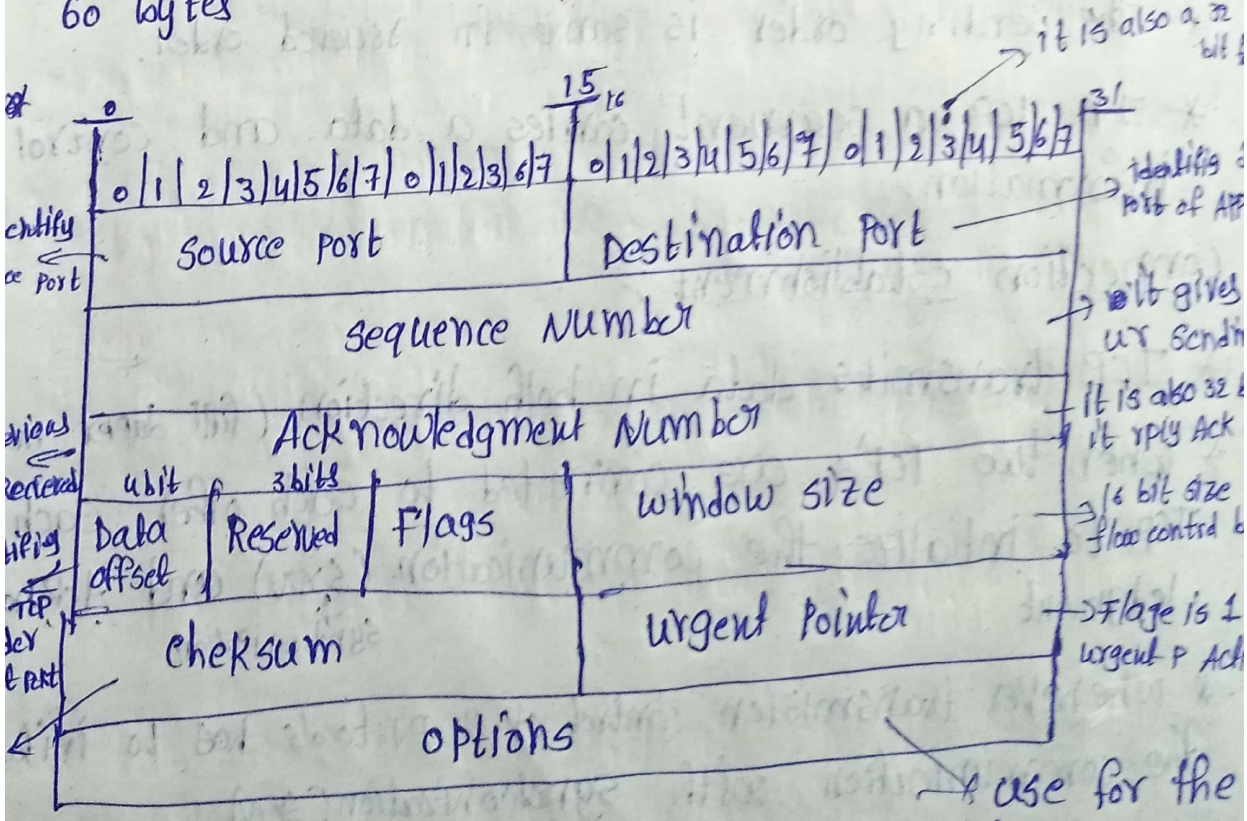
* your giving Ack and some information

* we called as Three way hand shaking protocol is used to establish connection b/w two TCP's



Header format of TCP

* The length of TCP header is min 20 bytes long & Max 60 bytes



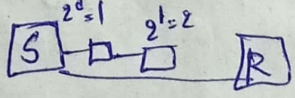
TCP Congestion control

* A congestion occurs, if the load offered to any network is more than its capability. → the congestion occurs.
* TCP controls congestion by means of window Mechanism
* TCP sets a window size telling the other end how much data segment to send.
* TCP window Mechanism is doing to telling the other end how much data segment to send. (window size)
* The TCP may use three algorithms for congestion control is
  1. slow start algorithm
  2. congestion avoidance algorithm
  3. congestion detection algorithm

1. Slow Start Algorithm

⇒ In this algorithm, the size of window grows exponentially till the time out occurs or the reciever's window is reached to the maximum threshold (64 kb = 65535 bytes)

Eg:- suppose sender starts with congestion window $= 2^0 = 1$ MSS



(max seg size, each segment consi of 1 byte)

⇒ After recieving a Ack for seg 1, the size of window can be increased by one.

∴ The total size of congestion window $= 2^1 = 2$

* when all seg Ack, total size of congestion window $2^3 = 8$

* So this is the one of the control of congestion algorithm.

2. Congestion Avoidance Algorithm

* In slow start alg, the size of congestion window increas exponentially

⟹ Here to avoid congestion before it happens, it is necessary to slow down the exponential growth

* How we can slow down the exponential growth?
  * The congestion avoidance alg works with the additive increase instead of exponential growth.
* In this algorithm, after receiving each ack reciept (for one round), the size of congestion window can be increased by one.

Eg: start with congestion window = 1
By adding 1, congestion window = 1+1 = 2
Nxt it becomes 3 like that.

Congestion detection Algorithm

* if the congestion occurs, it is required to derease the congestion on window size. → In slow starte exponentialy incre
  in congestion Avoidance increase by
  in conge detection decreas window

How can decreas?

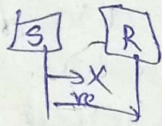
* There are two conditions in which congestion may occu
  The connection time out }
  the reception of three acknowledgment. } in those cases congestio may oc we detected.
* In both cases, the threshold size can be dropped to one half of the previous window size.

Quality of services (QoS)

QoS is an overall performance measure of computer Network

Flow characteristics of QoS are:-

1. Reliability
2. Delay
3. Jitter
4. Band width



Reliability

if a packet get lost (or) Ack not recieved

The sender s sends a packet is lost The sender retransmi the packet then reliability is decreased

* if packet lost, Ack lost the reliability is decreased.

Eg:- email & file transfer

② Delay

something happend blw source and destination is slo are not exct time it is Delay

* A Delay of source to destination have important character

Eg:- audio conference

Jitter

It is the variation in Packet delay. if the differe blw is larger it is high Jitter

* If the difference blw delay is smaller Then it i low Jitter.

Eg: 3 Packets sent at a time

① 0, 1, 2 → 10, 11, 12 → recieved time

Here delay is same to all 0-10, 1->10 also 10 delay

*So it is acceptable by telephone conversation.

② 0, 1, 2 ——→ 31, 34, 39

Here delay b/w the packets are different. So it is not acceptable by telephone conversation.

4 Bandwidth

* different applications need different bandwidth
* obiviously the applications are different the bandwidth also different

eg:- video conferencing.

congestion control Algorithm

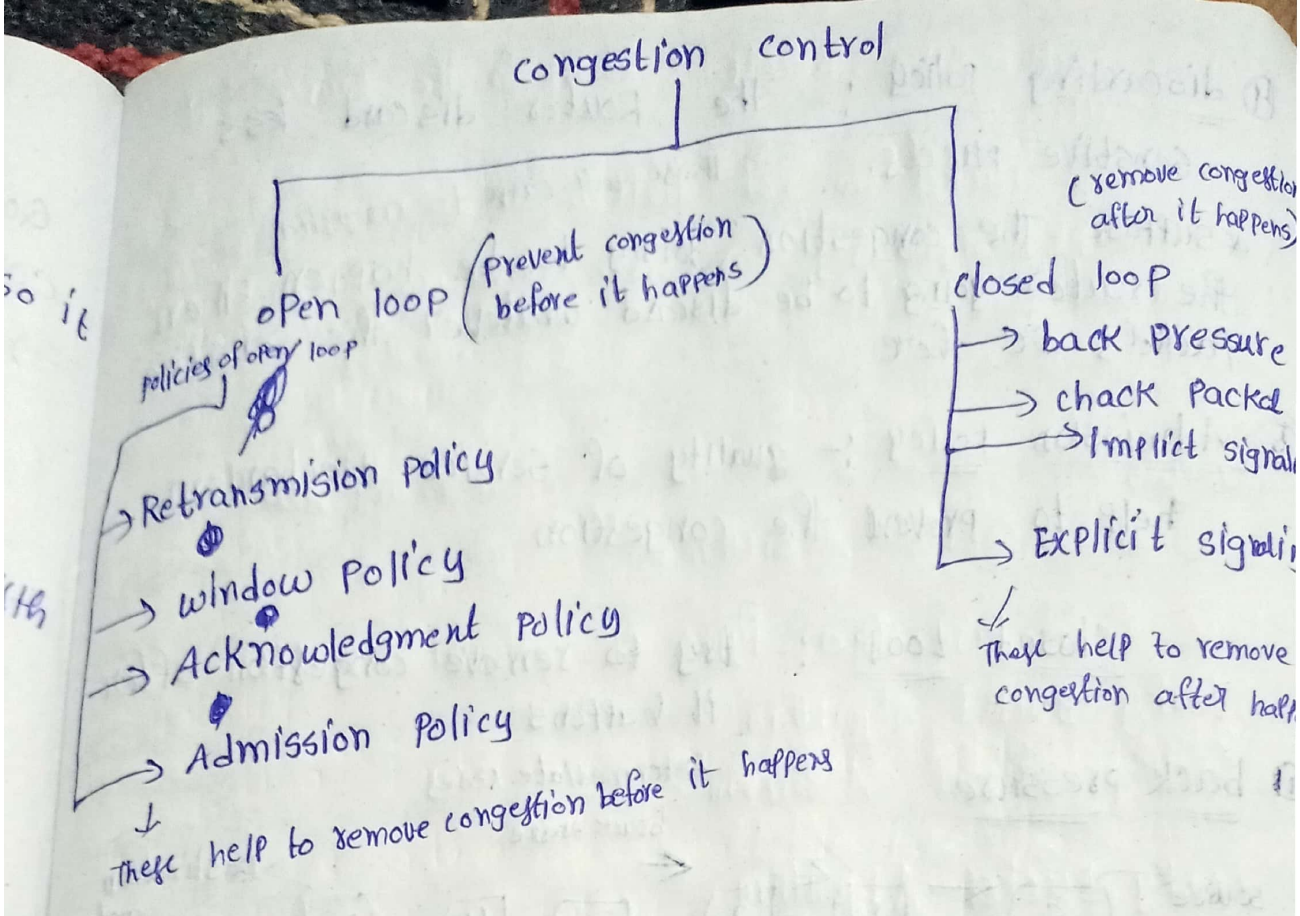
* congestion is a network may occur if the load of the N/w is great than capacity of n/w
* suppose in our home laptop, phn accessing same internet and downloads. There is lot of congestion over Network
* The downloading info(Movies) having more N/w than capacity of our N/w Then it decreases

Congestion control

congestion control refers to the mechanisms & techniques that can either prevent congestion before it happens or remove congestion after it happened.

Congestion control Mechanism is divided into the 2 types

* open loop
2) closed loop



open loop :- prevent congestion before it happens

① Retransmission :- Packet can be retransmit from sourc again

ex: if Packet sends the sender not recieve Ack. it assum the Packet not recieved by reciever. so it transmit again

② window Policy :- To implement window policy use selective reject window method for congestion contr with these technique help to contrd congest

③ Acknowledgment policy :- Reciever sending the Ack to send

R => Ack => S

* sender recieved Ack if assume what is sent the data is safely reached.

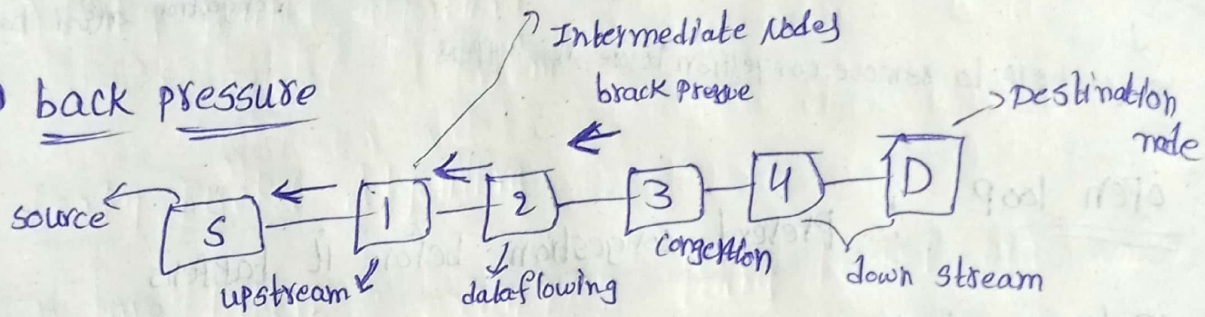
④ discarding policy :- The Router discard less sensitive packets. → these will cause to congestion

* suppose the congestion going to be happen. Then the router going to be discard whatever the sensitive packets are there

⑤ Admission policy :- quality of service mechanism help to prevent the congestion

Closed Loop :- try to remove congestion after it happens

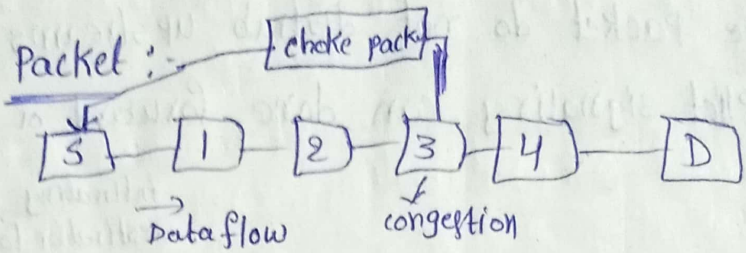
① back pressure



* so packet sends from source to destion.
* suddenly node a congestion is occur.
* The Node 3 will do is back pressure ← ← to the upstreams.
* whenever congestion occur at Node 3 it inform to the ups that is Node 2 to slow down the packet. it says slow down the packet congestion occur in my place
* Node 3 inform to Node 2 to use back pressure. Now No 2 congested and inform to Node 1. Nod1 congested and inform to source 5
* In this way the congestion is elivated.
* Node 3 inform to Node 2. Now Node 3 free and ready to frwrd N
* Node 2 inform to Node 1. Now Node 2 free and ready to transfer

* Node 1 inform to source. Node 1 free and transfer to Node 2.
* so this way congestion removed

② Choke Packet :-



* In Node3 congestion will occur. it directly send I choke Packet to the source
* It Node disturbing the upstream packets.
* ~~The~~ choke packet ~~to the source~~ with the information (c Node sending congestion occured in my node.
* so the soure can overcome this.

③ Implicit signaling

* It is simple concept. In Implict signaling there is No communication blw the congested Node and Sourc Node
so Here source guesses there is a congestion in netw when it does not recieve recieve any Ack from recie whenever congestion observed by source the source becom slow down.

Explicit signaling

it is Ntg but The congested node sending explic send a signal to the source ~~or~~ or destination.
- To inform about congestion but it is different fro the choke
- Informing or sending the direct signal to source or destination that congestion occured in intermediate Node

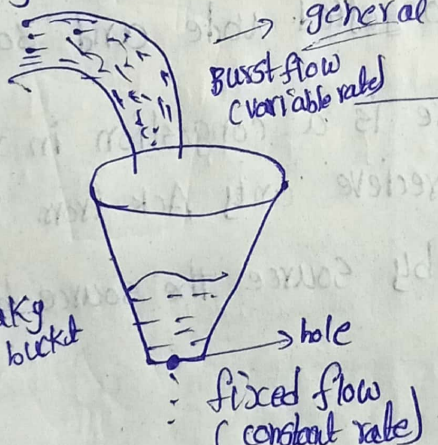
* choke packet also directly sending to the source. but it is different
* The choke Packet do not distrub upstreams intimating to source to sending Pkts
* The explict signaling can done forward or backward
  ↓
  intimating destination to slow down ~~sending packets~~ to sending Ack
* direction

~~cauges~~
congestion control Algorithms

Here there are two types of Algorithms
1) Leaky Bucket Algorithm
2) Token Bucket Algorithm

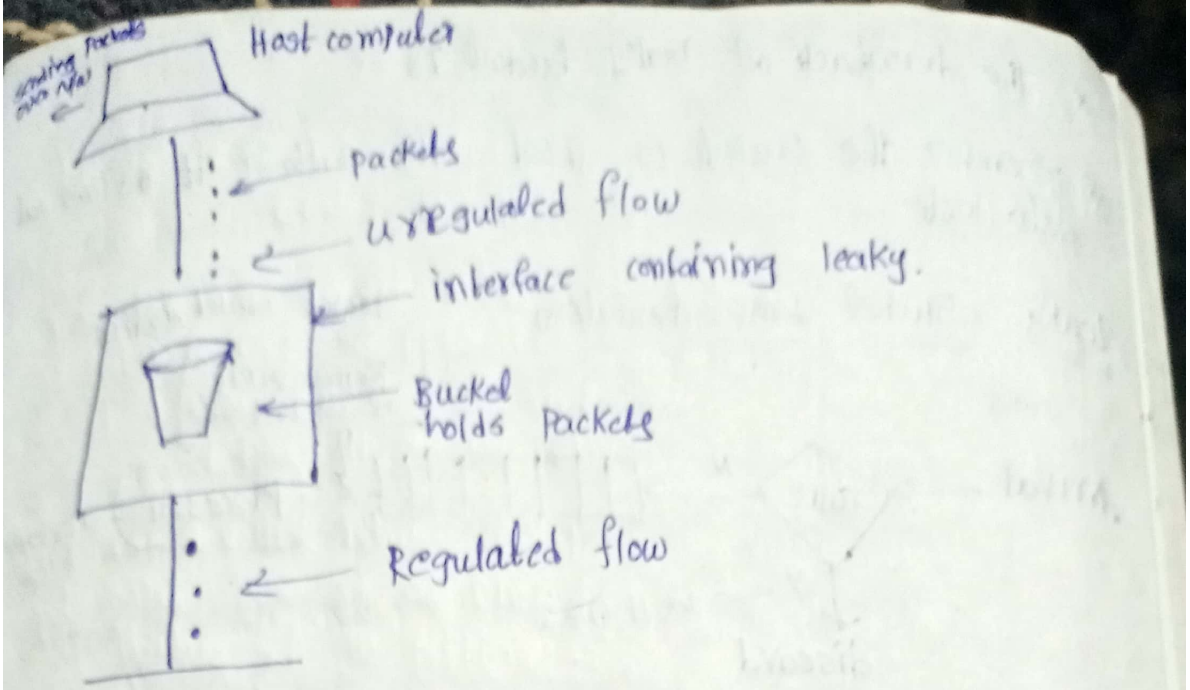
Leaky Bucket Algorithm

I have taken 2 examples Leaky bucket with water, leaky bucket with packets.



you are filled with water and hole on bucket. ~~bucket~~ but water gone slowly and poured fastly. So it filled out of bucket

* suppose pouring constant rate and coming out also constant rate no problem.
* but flow (variable rate) coming out is ~~fixed~~ (constant ra



* The compulor sends packet to the Bucket. sometime it sends 2 Packets sometime it 5 packets like that. so, it is unregulated flow.

* when host compulor send packet to bucket. The bucket have leaks at constant rate

* The bucket recieving not constantly. but leaking constantly rate

* The Host send irregular manner but bucket leakes at constant rate that is a Nw interface packets Regulated flow

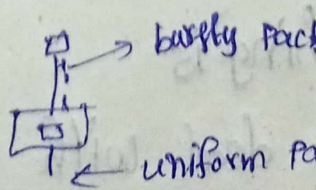
* It leaks 1 packet at a time. constant rate

Whenever congestion is there what will do?

* it holds the packet sends No matter how many packets sender is send.

* but it sends out to the 1 after another regulated flow

* So Here the bursty traffic converted to a uniform traffic by leaky bucket.

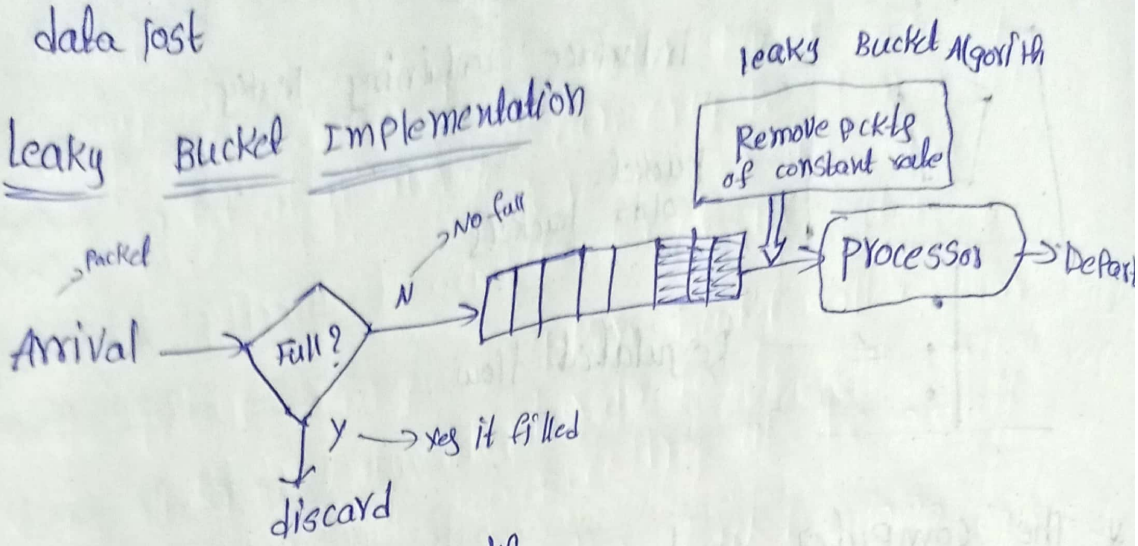


* So, this is the concept of leaky bucket.

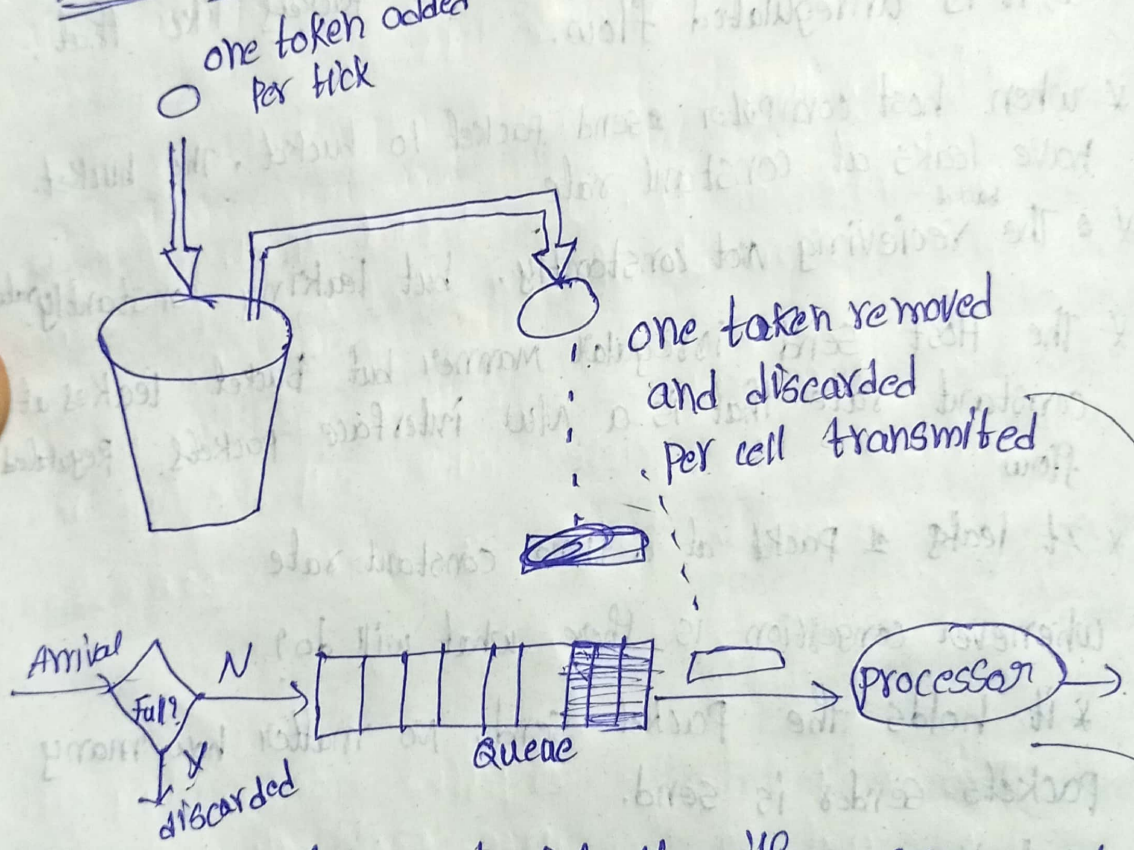
* The drawback of leaky bucket is

* Whenever the bucket is filled with packets it flows out data lost

Leaky Bucket Implementation



Token Bucket Algorithm



Here is the token bucket Algorithm same as the concept of leaky bucket.

but only thing is

Inorder to deal with the bursty traffic

that occurs in the leacky bucket we need a flexible algorithm. so that the data is not lost.

* The main concept of token bucket is the data is not lost.
* what it is doing. let see?
* In regular interaval tokens are thrown into a network (one token added per tic) → regulated flow
* one by one tokens are thrown into bucket.
* The bucket has a maximum capacity. The capicty will be alloted to the bucket
* whenever the bucket capacity is full ∀. it stops taking the tokens
* until the capacity is full. They hold the tokens and send those token into the 
* all the tokens are grouped  into Packet transfer to the destination.
* if there is a ready packet. a packet is remove from the bucket 
* suppose if No tokens in the ~~packet~~ bucket. The packet can not be sent

* Arrival it is Full discarded. otherwise (N) it sends to the queue one after another The packet will be sent to the processor and it will be departured.

Leaky Bucket vs Token Buckets

Leaky Bucket	Token Bucket
1) Leaky Bucket discard packets	1) Token Bucket does not, it discards tokens
2) ~~Token~~ Leaky Bucket ~~does not~~ sends packet at an average rate	Token Bucket ~~does not~~, a Packet can only be transmitted if there are enough tokens to cover its length in bytes
	Token Bucket allows for large bursts to be sent faster by speeding up the output
Leaky Bucket does not allow saving.	token Bucket allows saving up tokens (permission) to send ~~long~~ large bus