UNIT-I

1) Review of ISO OSI Reference Model
2) TCP/IP Architectures
3) Network Layer :-
   i) Design Issues
   ii) Services
   iii) Internal Organization
   iv) Comparison of Virtual circuits and Datagram Subnets
4) Routing Algorithms :
   i) The Optimality principle
   ii) Shortest path routing
   iii) Flooding
   iv) Flow-based algorithms
   v) Distance vector routing
   vi) Link State routing
   vii) Hierarchical routing
   viii) Broadcast & Multicast routings
   ix)
5) Congestion Control Algorithms :-
   i) General principles
   ii) Traffic Shaping
   iii) Congestion Control in Virtual circuit subnets
   iv) Choke packets
   v) Load Shedding
   vi) Jitter Control and Congestion control for Multicasting
   vii) Quality of Service (QOS)

# NETWORK LAYER



## Introduction :-

→ Lowest Layer that deals with end-to-end transmission

→ Concerned with getting packets from the source all the way to the destination.

→ Delivering packets to the destination may require making many hops at intermediate routers along the way.

→ To achieve the above goals,

1) N/w layer must know about the topology of the communication subnet (i.e., set of all routers).
2) Select appropriate paths to avoid overloading
3) Handle problem faced when source and destination are on different networks.

## Network Layer Design Issues :-

→ Network Layer Issues Includes,

1) The service provided to the transport layer
2) The Internal design of the subnet.

→ Design Issues are :-

1) Store-and-Forward Packet Switching
2) Services provided to the Transport Layer
3) Implementation of Connectionless Service
4) Implementation of Connection-Oriented Service
5) Comparison of Virtual-Circuit and Datagram Subnets

1) Store-and-Forward Packet Switching :-

→ Major components of the system are the carrier's equipment (routers connected by transmission lines).

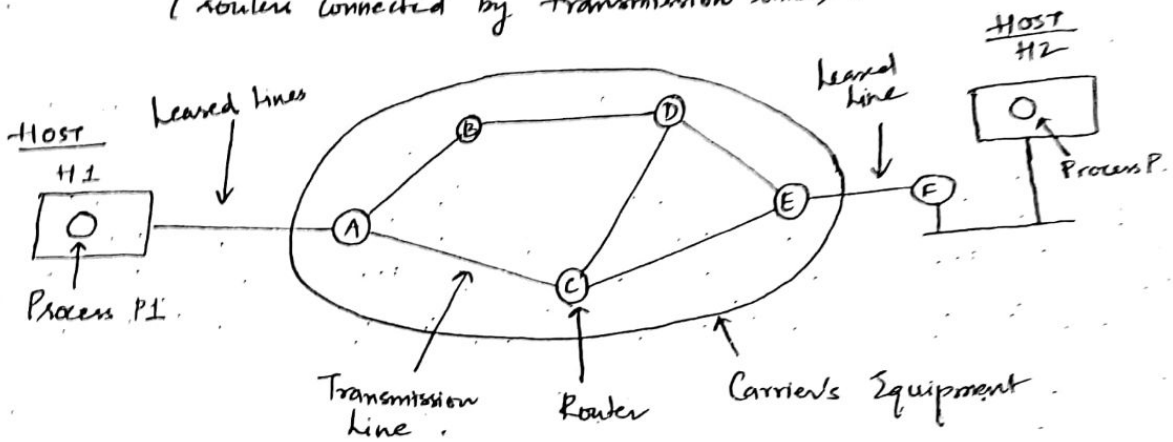



Fig: The Environment of the Network Layer Protocols.

→ Host H1 is directly connected to one of the carrier's router A by a leased line

→ Host H2 is on a LAN with a router, F owned and operated by the customer

→ A host with a packet sends it to the nearest router, either on its own LAN or over a point-to-point link to the carrier.

→ The packet is stored there until it has fully arrived, so that the checksum can be verified. Then it is forwarded ~~the~~ to the next router along the path until it reaches the destination host, where it is delivered. This mechanism is "Store-and-Forward Packet Switching".

2) Services Provided to the Transport Layer :-

→ Network layer provides services to transport layer at the Network layer / Transport layer interface.

→ Goals of Network layer services :-

1) Services should be independent of the router technology
2) Transport layer should be shielded from the number, type and topology of the routers present.
3) The network addresses made available to the transport layer should use a uniform plan, even across LAN's and WAN's.

→ Network layer can be Connection-oriented or Connectionless (Independent of transport layer).

→ Internet community suggest subnet is inherently unreliable no matter how it is designed. ∴ this suggest network services should be Connectionless.

→ Telephone companies suggest subnet should provide a reliable connection-oriented service. Since QoS is the main factor and without connections in Subnet, it is very difficult to achieve.

→ These two camps are best exemplified by the,
i) Internet :- Connectionless Network layer
ii) ATM :- Connection Oriented Network layer

b) Implementation of Connectionless Service :-

→ Packets are "injected into the subnet individually and routed independently of each other. No advance step is needed.

→ Such packets are frequently called "Datagrams" and subnet is called a "Datagram Subnet".

→ Datagram Subnet Working :-

Suppose the process P1 has a long message for P2.

1) Transport layer code prepends a transport header to the front of the message and hands the result to the Network layer.

2) Assume, message is 4 times longer and broken into 4 packets 1, 2, 3 and 4 and sends each of them in turn to router A using some point-to-point protocol (Eg. PPP)

3) Every router has an internal table telling it where to send packets for each possible destination. Each table entry is a pair consisting of a destination and the outgoing line to use for that destination.

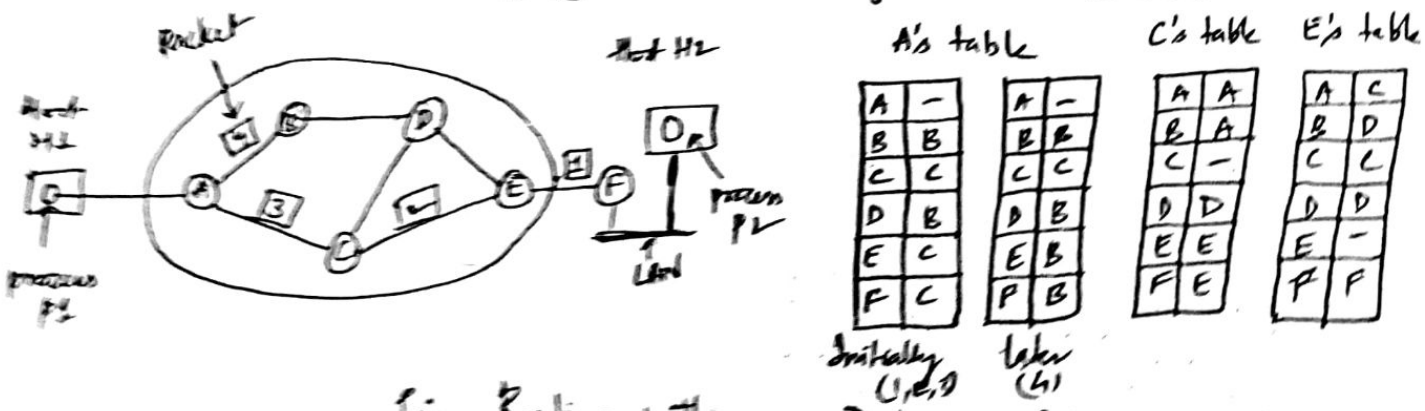


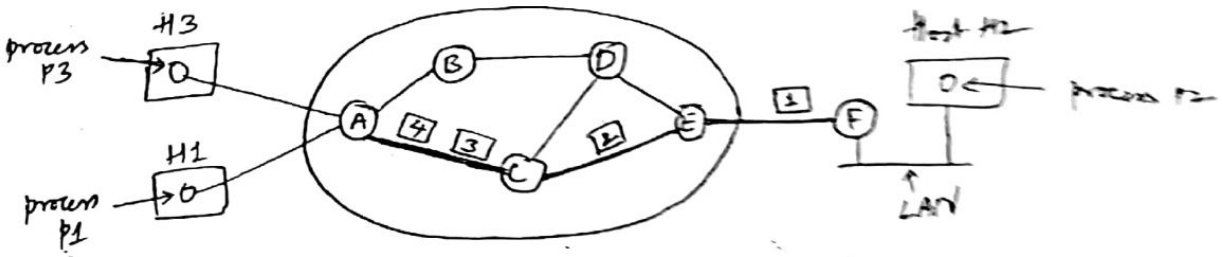

Fig:- Routing within a Datagram Subnet

## Implementation of Connection Oriented Service :-

→ Connection oriented service requires a path to be established from source router to destination router before any packets can be sent. This connection is called a "Virtual Circuit (VC)" and the subnet is called "Virtual Circuit Subnet".

→ Avoids having to choose a new route for every packet sent.

→ For example, Host H1 has established connection 1 with Host H2. It is remembered as the first entry in each of the routing tables




A's Table

| In | | Out | |
|---|---|---|---|
| H1 | 1 | C | 1 |
| H3 | 1 | C | 2 |

C's Table

| In | | Out | |
|---|---|---|---|
| A | 1 | E | 1 |
| A | 2 | E | 2 |

E's Table

| In | | Out | |
|---|---|---|---|
| C | 1 | F | 1 |
| C | 2 | F | 2 |

fig :- Routing within a Virtual-circuit Subnet.

→ When H3 establishes a connection to H2, this leads to the second entry in the routing tables.

→ H3 uses the same connection "connection-1" but router A assigns a different connection identifier (2) to the outgoing traffic for the second connection. This allow routers C & E identify that packets are received from Host H3.

⑤ Comparison of Virtual-Circuit and Datagram Subnets :-

| Sno | Issue | Datagram Subnet | Virtual-Circuit Subnet |
|---|---|---|---|
| 1 | Addressing | Each packet contains the full source and destination address. | Each packets contains a short VC number |
| 2 | Circuit Setup | Not needed | Required |
| 3 | State Information | Routers do not hold state information about connections | Each VC requires router table space per connection |
| 4 | Routing | Each packet is routed independently | Route chosen when VC is setup : All packets follow it |
| 5 | Effect of router failure | None, Except for packets lost during the crash | All VC's that passed through the failed router are terminated |
| 6 | Quality of Service | Difficult | Easy if enough resources can be allocated in advance for each VC |
| 7 | Congestion Control | Difficult | Easy if enough resources can be allocated in advance for each VC |

Fig :- Comparison of datagram and Virtual-circuit subnets.

## Routing Algorithms :-

→ The main function of the network layer is routing packets from source machine to destination machine.

→ Routing algorithms is responsible for,

i) looking up the routing tables to decide the outgoing line for each packet that arrives. Also known as "Forwarding".

ii) Filling in and updating the routing tables.

→ Routing algorithm is that part of network layer software responsible for deciding which output line an incoming packet should be transmitted on.

(or)

→ The algorithm that manages the tables and makes the routing decisions is called the routing algorithm.

→ Routing algorithms can be grouped into two major classes :-

1) Non-Adaptive Algorithms - (Static Routing)

2) Adaptive Algorithms - (Dynamic Routing)

1) Non-Adaptive Algorithms :-

→ Routing decisions are not based on measurements / estimates of current traffic and topology

→ Choice of route to use is computed in advance, offline and downloaded to routers when the network is booted.

2) Adaptive Routing :-

→ changes routing decisions to reflect changes in topology and usually traffic as well.

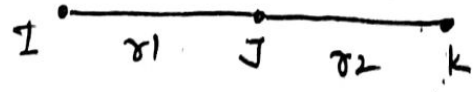
→ Adaptive algorithm differs in

1) Where they get their info – { locally from adjacent routers (or) From all routers }

2) When they change routes – { Every ΔT secs, (or) When load changes (or) when topology changes }

3) What metric is used for Optimization – { e.g.: Distance, (or) No. of hops (or) Estimated transit time }

The Optimality Principle :- It states that if router J is on the optimal path from router I to router k, then the optimal path from router J to router k also falls along the same route.

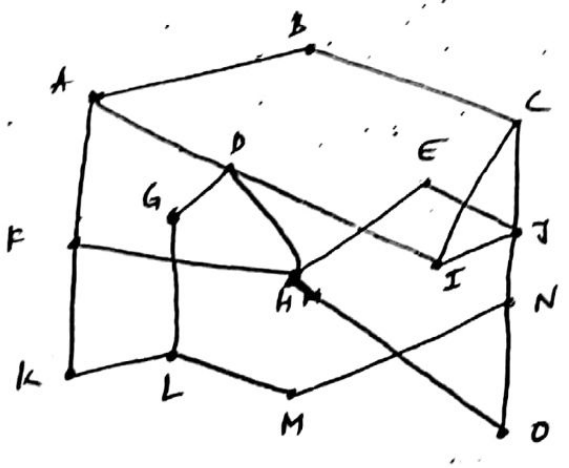


→ Direct consequence of optimality principle is that we can see a set of optimal routes from all sources to a given destination forming a tree rooted at the destination. Such a tree is called a "Sink Tree".

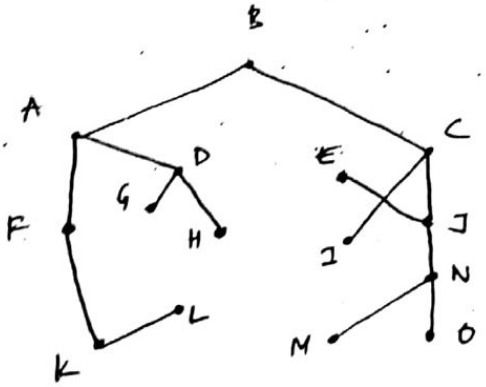
→ Sink Tree is not necessarily unique, other trees with same path length may exists.

→ The goal of all routing algorithms is to discover and use the sink trees for all routers.

→ Optimality principle and the sink tree provides a benchmark against which other routing algorithms can be measured.




(a) Subnet




(b) Sink Tree for router B

## Shortest Path Routing :-

→ Subnet is a graph, with each node of the graph representing a router and each arc of the graph representing a communication line (often called links).

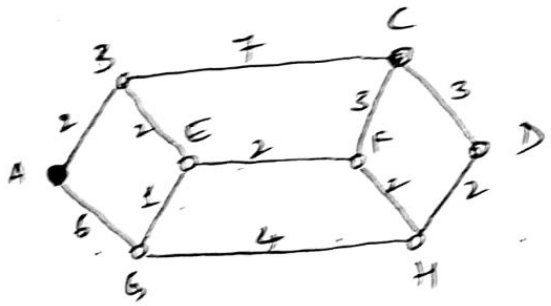
→ The algorithm just finds the shortest path between a given pair of vertices in the graph.

→ In general, the labels on the arcs could be computed as a function of the :

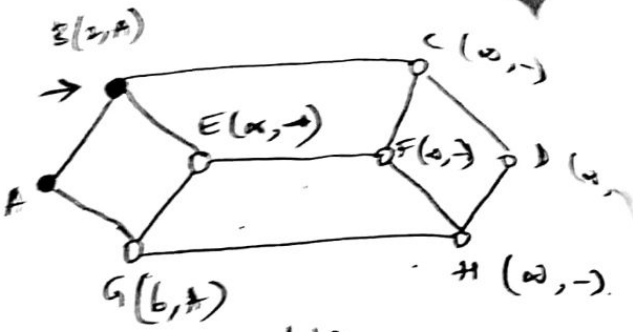
i) distance
ii) Bandwidth
iii) Average Traffic
iv) Communication Cost
v) Mean Queue Length
vi) Measured Delay

} Metrics for computing shortest path

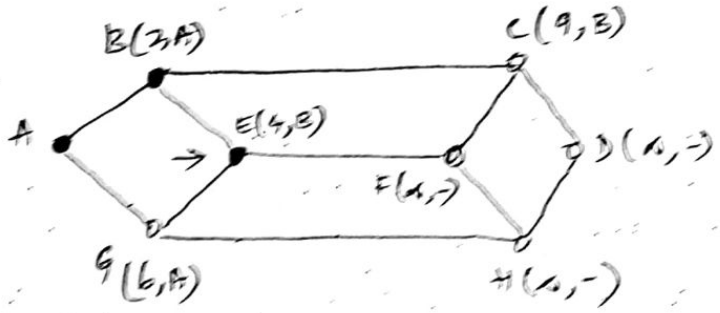
→ Dijkstra's shortest path algorithm can be used to compute shortest path between two nodes of a graph.

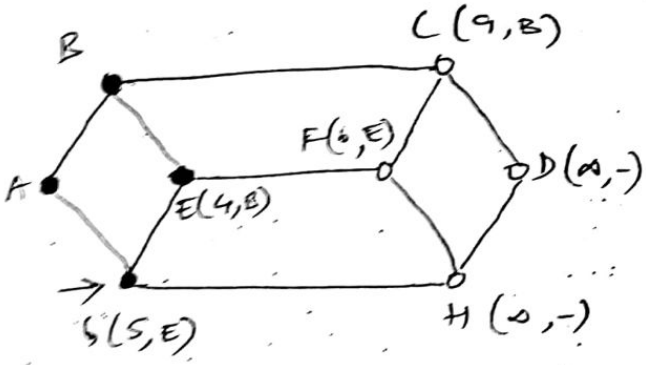


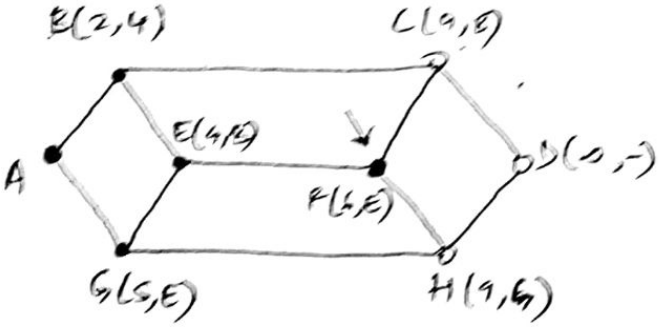

(a)




(b)



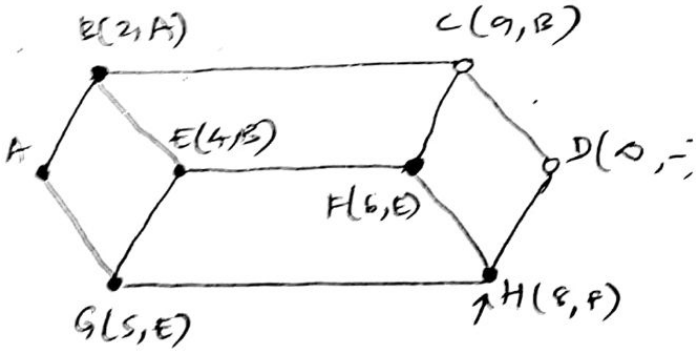




Fig:- 5 steps used in computing shortest path from A to D.

→ Initially, all labels are tentative (○).

→ ~~One node is selected as the path from the source to node is established~~

→ Once the label represents the shortest possible path from the source to that node, it is made permanent (●) and never change thereafter.

## Flooding :-

→ A static routing algorithm in which every incoming packet is sent out on every outgoing line except the one it arrived on.

→ Generates vast number of duplicate packets, in fact infinite number unless some measures are taken to damp the process.

→ Hop Counter :-

i) Hop Counter can be stored in each packet header.

ii) Hop Counter gets decremented at each hop, with the packet being discarded when Counter reaches zero.

⇒ Requires, hop counter to be initialized to the length of the path from src to dest

⇒ In worst case, it can be initialized to the diameter of the subnet.

→ Keeping track of Packets that have been flooded :-

i) The source router put a sequence number in each packet it receives from its hosts.

ii) Each router maintains a list for each source router telling which sequence numbers originating at that source have already been seen.

iii) If an incoming packet is on the list, it is not flooded.

⇒ To prevent the list from growing without bound, each list should be augmented by a counter k, meaning that all sequence numbers through k have been seen.

→ Selective Flooding :-

A variation of flooding in which routers send incoming packet out on those lines that are going approximately in the right direction.

→ Flooding Applications :-

⇒ Military application in which tremendous robustness of flooding is highly desirable.

⇒ Distributed Database applications, sometimes requires to update all databases concurrently.

⇒ In Wireless N/w's where all messages transmitted by a station can be received by all the other stations within its radio range.

## Distance Vector Routing :-

→ A dynamic routing algorithm, which operates by having each router maintain a table (i.e, a vector) giving the best known distance to each destination and which line to use to get there.

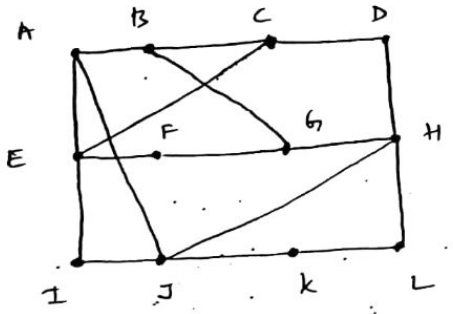
→ It is Also called Bellman-Ford routing algorithm and the Ford-Fulkerson algorithm after the researchers who developed it.

→ In Distance Vector Routing, Each router maintains a routing table that contains two parts :
   i) Preferred outgoing line to use for that destination
   ii) Estimate of time/distance (or any other metrics).

→ For example, Assume delay is used as metric, and once every T msec each router sends each neighbour a list of its estimated delays to each destination.

→ Assume X receive such a table from its neighbour, where $X_i$ is the delay to get to the router $i$.
Let "m" be the delay from X to that neighbour router
∴ Time delay from X to router $i = \boxed{X_i + m}$ msec.

→ Suppose "J" has measured or estimated its delay to its neighbours A, I, H and K as 8, 10, 12 and 6 msec, respectively.




| To | A | I | H | K | New Estimated delay | New Routing Table for J |
|---|---|---|---|---|---|---|
| A | 0 | 24 | 20 | 21 | 8 | A |
| B | 12 | 36 | 31 | 28 | 20 | A |
| C | 25 | 18 | 19 | 36 | 28 | I |
| D | 40 | 27 | 8 | 24 | 20 | H |
| E | 14 | 7 | 30 | 22 | 17 | I |
| F | 23 | 20 | 19 | 40 | 30 | I |
| G | 18 | 31 | 6 | 31 | 18 | H |
| H | 17 | 20 | 0 | 19 | 12 | H |
| I | 21 | 0 | 14 | 22 | 10 | I |
| J | 9 | 11 | 7 | 10 | 0 | — |
| K | 24 | 22 | 22 | 0 | 6 | K |
| L | 29 | 33 | 9 | 9 | 15 | K |
| | JA delay is 8 | JI delay is 10 | JH delay is 12 | JK delay is 6 | | |

Vectors received from J's neighbours

→ J computes its new route to router G as :

$= \min\{(JA+AG), (JI+IG), (JH+HG), (Jk+kG)\}$

$= \min\{(8+18), (10+31), (12+6), (6+31)\}$

$= \min\{26, 41, 18, 37\}$

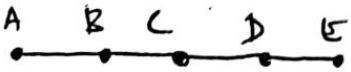
$= 18$ from router H.

→ Drawback of Distance Vector Routing algorithm is that it reacts rapidly to good news, but leisurely to bad news.


| | | | | |
|---|---|---|---|---|
| . | . | . | . | Initially |
| 1 | . | . | . | After 1 exchange |
| 1 | 2 | . | . | " 2 " |
| 1 | 2 | 3 | . | " 3 " |
| 1 | 2 | 3 | 4 | " 4 " |

Good news

(Metric = No. of hops)




Bad news (Link B to A goes down)

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | Initially |
| 3 | 2 | 3 | 4 | After 1 exchange |
| 3 | 4 | 3 | 4 | . 2 , |
| 5 | 4 | 5 | 4 | . 3 . |
| 5 | 6 | 5 | 6 | . 4 " |
| 7 | 6 | 7 | 6 | . 5 " |
| 7 | 8 | 7 | 8 | " 6 " |
| . | . | . | | |
| . | . | | | |
| . | | | | |

Ex:- Count-to-Infinity problem

## Link State Routing :-

→ A Dynamic routing algorithm.

→ Link state routing can be stated as 5 parts:

1) Discover its neighbours and Learn their N/w addresses.
2) Measure the delay or cost to each of its neighbours
3) Construct a packet telling all it has just learned.
4) Send this packet to all other routers.
5) Compute the shortest path to every router.

→ Complete topology and all delays are experimentally measured and distributed to every router. Then the Dijkstra's algorithm can be run to find the shortest path to every other router.

### (1) Learning about the neighbors :-

→ When a router is booted, its first task is to learn who its neighbours are.

→ This is accomplished by sending a special "HELLO" packet on each point-to-point line

→ The router on the other end is expected to reply back telling who it is.

Measuring Line Cost :-

→ Each router must have a reasonable estimate of the delay to each of its neighbours.

→ The most direct way to determine this delay is to send over line a special "ECHO" packet that the other side is required to send back immediately.

→ Measuring the round trip time and dividing it by two gives sender a reasonable estimate of delay.

→ An important issue is whether to consider the load in measuring delay or not.

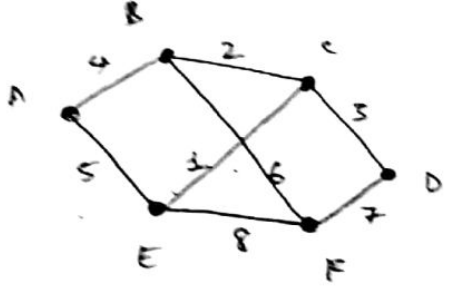
→ If load is considered then it is better to distribute the load over multiple lines, with some known fraction going over each line

Building Link State Packets :-

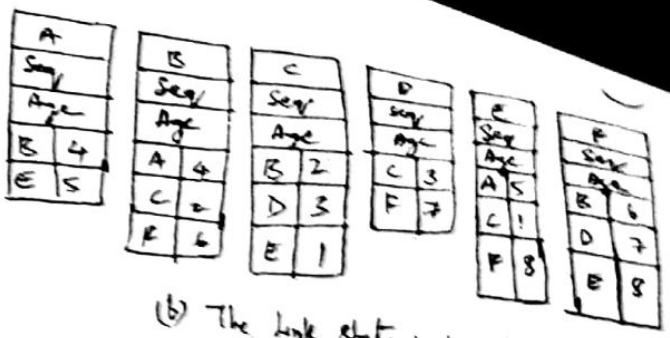
→ Each router needs to build a packet containg all the data

→ The packet starts with the

i) Identity of sender
ii) Sequence Number
iii) Age
iv) List of neighbours (Neighbour Node, Delay).




(a) Subnet




(b) The link state packets for subnet (a)

→ Building link state packets is easy. The crucial part is determining when to build them.

→ One solution is to build them periodically at regular intervals or another solution is to build them when some significant event occurs (like a line / neighbour goes down or comes back or changes its properties appreciably).

Distributing the Link state Packets :-

→ The fundamental idea to distribute the link state packets is to use flooding.

→ To keep the flood in check, each packet contains a sequence number that is incremented for each new packet sent.

→ Routers keep track of all the (source router, sequence) pairs they see.

→ When a link state packet arrives at a router, it is checked against the list of packets already seen,

i) If it is new, it is forwarded on all lines except the one it arrived on.

ii) If it is a duplicate, it is discarded.

iii) If its sequence number is lower than it is rejected

problems that can arise :-

1) Sequences numbers may wrap around.
2) If a router crash, it will lose track of its sequences number. { start again at 0
3) Sequence number may get corrupted. { from 4 to 65,540 (A 1 bit error) packets 5 to 65,540 will be rejected.

Solutions :-

1) Use a 32-bit sequence number. With one link state packet per second, it would take 137 years to wrap around.

2&3) Solution to these problems is to include the age of packet after the sequence number and decrement it once per second. When the age hits zero, the information from that router is discarded.

⇒ If the age of packet decrements once per second, and say every 10 sec a new packet comes in, so router information times out when a router is down or six consecutive packets have been lost.

⇒ To guard against errors on router - router lines, all link state packets are acknowledge.

⇒ Data structure used by router B for the subnet records source, sequence number and age, and the data. In addition there are send and acknowledge flags for each of B's three lines (to A, C, and F respectively).

⇒ send line [redacted] means that the packet must be sent on the indicated line

→ A.... 1.1.. 0. 1... means that it must be ack there

| | | | Send flag's | | | ACK Flags | | | |
|---|---|---|---|---|---|---|---|---|---|
| Source | Seq | Age | A | C | F | A | C | F | Data |
| A | 21 | 60 | 0 | 1 | 1 | 1 | 0 | 0 | |
| F | 21 | 60 | 1 | 1 | 0 | 0 | 0 | 1 | |
| E | 21 | 59 | 0 | 1 | 0 | 1 | 0 | 1 | |
| C | 20 | 60 | 1 | 0 | 1 | 0 | 1 | 0 | |
| D | 21 | 59 | 1 | 0 | 0 | 0 | 1 | 1 | |

fig:- the packet buffer for router B

⑤ Computing New Routes :-

→ Once a router has accumulated a full set of link state packets, it can construct the entire subnet graph, because every link is represented.

→ Every link is represented twice, once for each direction. The two values can be averaged or used separately.

→ Dijkstra's algorithm can be run locally to construct the shortest path to all possible destinations. The result of this algorithm can be installed in the routing tables, and normal operation resumed.

→ OSPF (open shortest path first) protocol and IS-IS (Intermediate System - Intermediate System) are example protocols that uses link state protocol.

## Hierarchical Routing :-

→ As the network grow in size, the routing tables grow proportionally.

→ Increase in network size results in,

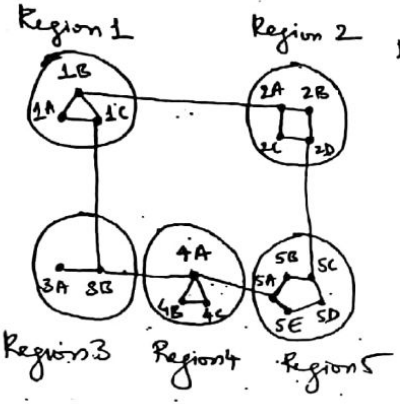
1) Increased in router-memory
2) More CPU time needed to scan them
3) More bandwidth is needed to send status reports

→ Increase in network size may grow to a point where it is not feasible for every router to have an entry for every other router. In such a case routing must be done hierarchically.

→ Hierarchical routing, divides the routers into regions, with each router knowing all the details about how to route packets to destinations within its own region, but not knowing anything about the internal structures of other regions.

→ For huge networks, A two-level hierarchy may be insufficient; it may be necessary to group the regions into clusters, the clusters into zones, the zones into groups and so on until we run out of names for aggregations.

→ The below example gives presents a two-level hierarchy with five regions.




Full table for 1A.

| Dest | Line | Hops |
|---|---|---|
| 1A | — | — |
| 1B | 1B | 1 |
| 1C | 1C | 1 |
| 2A | 1B | 2 |
| 2B | 1B | 3 |
| 2C | 1B | 3 |
| 2D | 1B | 4 |
| 3A | 1C | 3 |
| 3B | 1C | 2 |
| 4A | 1C | 3 |
| 4B | 1C | 4 |
| 4C | 1C | 4 |
| 5A | 1C | 4 |
| 5B | 1C | 5 |
| 5C | 1B | 5 |
| 5D | 1C | 6 |
| 5E | 1C | 5 |

→ Contains 17 entries

Hierarchical table for 1A

| Dest | line | Hops |
|---|---|---|
| 1A | — | — |
| 1B | 1B | 1 |
| 1C | 1C | 1 |
| 2 | 1B | 2 |
| 3 | 1C | 2 |
| 4 | 1C | 3 |
| 5 | 1C | 4 |

↳ Contains 7 entries

Congest
1) Ger
2) C
3)
4)
5)
6)

due

m

fig:- Hierarchical Routing

→ The disadvantage of hierarchical routing is the increased path length. For example, the best route from 1A to 5C is via region 2, but with hierarchical routing all traffic to region 5 goes via region 3, because that is the better for most destinations in region 5.

→ If there are 720 routers in a subnet. then,

1) No hierarchy — Each router needs 720 entries.

two-level 2) 24 regions with 30 routers — Each router has 30 (local entries) + 23 (regions) = 53 entri

three-level 3) 8 clusters, 9 regions of 10 routers — Each router has 10 (local) + 8 (region) + 7 (cluster) = 25 ent

→ The optimal number of levels for an N router subnet is ln N, requiring a total of e ln N entries per router. (Discovered by Kamoun and Kleinrock (1979))

## Broadcast Routing:-

→ Sending packets to all destinations simultaneously is called Broadcasting.

For example, A service distributing weather reports, stock market updates or live radio programs works best by broadcasting to all machines.

→ The various methods proposed for broadcasting are:

~~i) Source sends packets~~

⇒ (1) Source simply send a distinct packets to each destination.

Disadvantages:-

(i) This method is wasteful of bandwidth.
(ii) Requires the source to have a complete list of all destinations.

→ (2) Flooding is another obvious choice.

Disadvantage:-

i) Generates too many packets
ii) Consumes too much bandwidth.

→ (3) Multidestination Routing: In this method, each packet contains either a list of destinations or a bitmap indicating the desired destinations.

⇒ When a packet arrives at a router,

i) the router checks all the destinations to determine the set of output lines that will be needed.

ii) The router generates a new copy of the packet for each output line to be used and includes in each packet only those destinations that are to use the line (i.e, destination set is partitioned among the output lines).

→ Multidestination routing is like separately addressed packets, except that when several packets must follow the same route, one of them pays full fare and the rest ride free.

→ (4) ⇒ Spanning Tree :- Algorithm makes explicit use of the sink tree ~~initiating~~ for router initiating the broadcast - or any other convenient spanning tree.

⇒ A spanning tree is a subset of the subnet that includes all the routers but contains no loops.

Advantage :-

i) This method makes excellent use of bandwidth, generating the absolute min^m no. of packets to do the job

Disadvantage:-

i) Each router must have knowledge of some spanning tree for the method to be applicable.

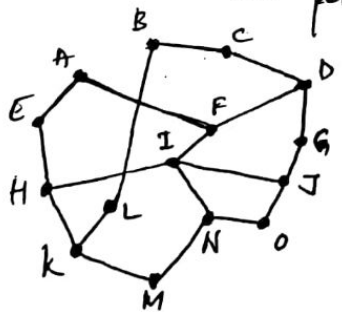
this information is available with link state routing but sometimes it is not (with distance vector routing).

→ (5) Reverse path forwarding:-

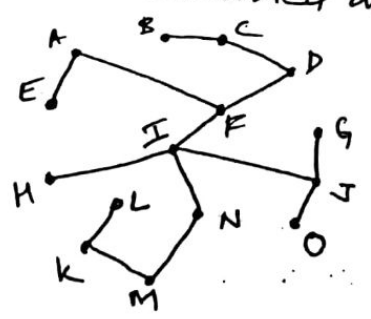
→ When a broadcast packet arrives at a router, the router checks to see if the packet arrived on the line that is normally used for sending packets to the source of the broadcast.

(i) ⇒ If so, there is an excellent chance that the broadcast packet itself followed the best route from the router and is therefore the first copy to arrive at the router. Thus, the router forwards copies of it on all lines except the one it arrived on

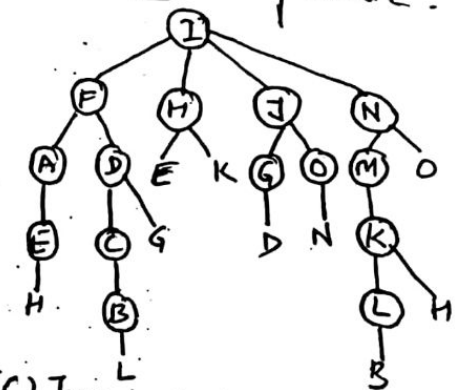
(ii) If the broadcast packet arrive on a line other than the preferred one for reaching the source, the packet is discarded as a likely duplicate.




(a) Subnet




(b) A Sink Tree




(C) Tree built by reverse path

→ After 5 hops and 24 packets, the broadcasting terminates, Compared with 4 hops and 14 packets had the sink tree been followed exactly.

Advantage :-

i) Reasonably efficient and easy to implement.
ii) Doesn't require routers to know about spanning trees.
iii) Doesn't have the overhead of a destination list or bit map in each broadcast packet
iv) Doesn't require any special mechanism to stop the process

## Multicast Routing :-

→ Some applications require that widely-separated processes work together in groups.
for eg., A group of processes implementing a distributed date base system.

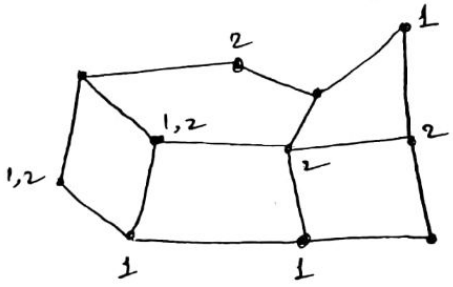
→ Sending a message to such a group is called multicasting and its routing algorithm is called multicast routing.

→ Multicasting requires group management. There is some way needed to create and destroy groups and to allow processes to join and leave groups

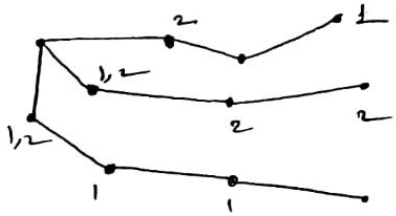
→ Finally, the routers must learn about which of their hosts are in which groups. Routers tell their neighbours, so the information propagates through the subnet.

→ To do multicast routing, each router computes a spanning tree covering all other routers.

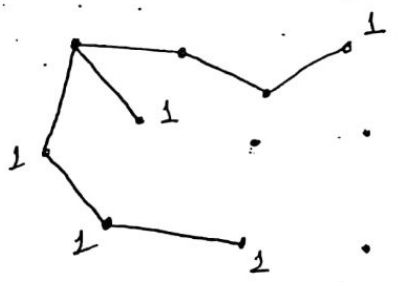
for example, consider we have two groups, 1 and 2. some routers are attached to hosts that belong to one or both of these groups as shown below.



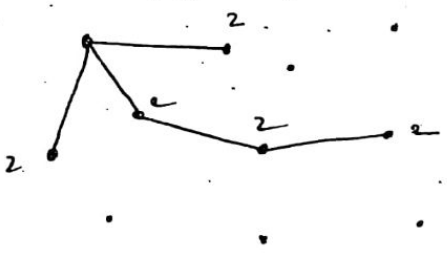

(a) A Network




(b) A spanning tree for the leftmost router




(c) A Multicast tree for group 1




(d) A multicast tree for group 2

→ When a process sends a multicast packet to a group, the first router examines its spanning tree and prunes it, removing all lines that do not lead to hosts that are members of the group.

# Congestion Control Algorithms

1) General principles of congestion control
2) Congestion Prevention policies
3) Congestion control in Virtual-Circuit Subnets
4) Congestion Control in Datagram Subnets
5) Load shedding
6) Jitter Control

## Introduction :-

→ Congestion is a situation where in too many packets are present in (a part of) the subnet, degrading its performance

→

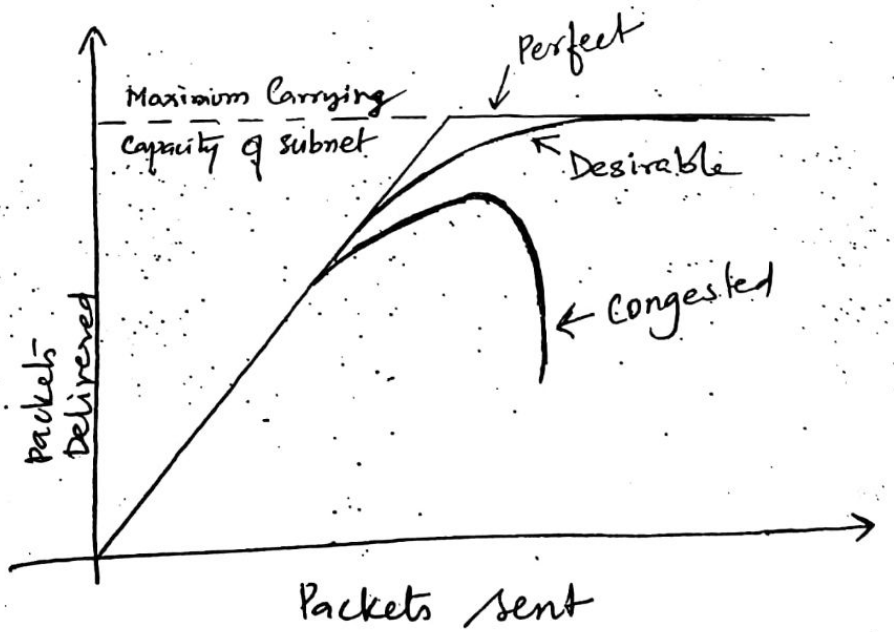



fig :- Performance degrades sharply in Congestion

Symptoms :-

→ When the no. of packets dumped into the subnet by the hosts is within its carrying capacity, they are all delivered (except few due to transmission errors) and the number delivered is proportional to the number sent.

→ As traffic increases too far, performance collapses completely and almost no packets are delivered.

Reasons / Causes :-

Congestion may occur due to the following reasons :

→ 1) Insufficient memory to hold packets at routers :-

⇒ If all of a sudden, streams of packets begin arriving on three or four input lines at a router and all need the same output line, A queue will be build up to store all the arriving packets but due to insufficient memory at routers results in packets lost.

⇒ Increasing memory at routers may result in Congestion getting worse, because the packets may time out (repeatedly) by the time they get to the front of the queue. resulting in duplicates to be sent.

2) Slow processing units at routers :-

⇒ If routers CPU are slow at performing bookkeeping tasks required of them like queuing buffers, updating tables etc., then queues can build up even though there is excess line capacity

3) Low-bandwidth lines can also cause congestion.

→ Upgrading a part, but not all, of the system often moves the bottleneck somewhere else and problem will persist until all the components are in balance.

## General Principles of Congestion Control :-

The Presence of congestion means that the load is (temporarily) greater than the resources (in part of system) can handle.

→ The two solutions to this problem is :

i) Increase the resources, or

ii) Decrease the load.

→ Increasing the resources :-

i) Subnet may leased lines temporarily to increase the bandwidth between certain points.

ii) Splitting traffic over multiple routes instead of always using the best route.

iii) Spare routers normally used as backups can be put online to give more capacity to subnet

→ Decreasing the load :-

⇒ If subnet capacity is increased to the limit, and if still congestion exists then the only way to deal with it is to decrease the load.

⇒ The following ways can be used to reduce load :

i) Denying service to some users

ii) Degrading service to some or all users.

iii) Having users schedule their demands in a more predictable way

→ For subnets using virtual-circuits, these methods can be used at the network layer.

→ For Datagram subnets, they can sometimes also be applied on transport layer connections.

→ The approaches to handle congestion can be divided into two groups:

i) Open loop
ii) Closed loop.

→ Open loop :-

⇒ Solution attempt to solve the problem by good design, in essence, to make sure it doesn't occur at first place.

⇒ Once the system is up and running, midcourse corrections are not made.

⇒ Open loop solutions make decisions without regard to the current state of the network

⇒ Tools for doing open loop control include:

i) Deciding when to accept new traffic
ii) Deciding when to discard packets
iii) Making scheduling decisions at various points in the network.

→ Closed loop :-

⇒ Closed loop solutions are based on the concept of a feedback loop.

⇒ Closed loop approach has 3 parts to control congestion

i) Monitor the system to detect when and where congestion occurs.

ii) Pass this information to places where action can be taken.

iii) Adjust system operation to correct the problem.

→ A variety of metrics can be used to monitor the subnet for congestion. Some of the important one are:

i) Percentage of all packets discarded due to lack of buffer space

ii) Average queue lengths

iii) Number of packets that time out and retransmitted

iv) Average packet delay

v) Standard deviation of packet delay

Increasing numbers in above metrics indicates growing congestion.

→ Closed loops algorithms are also divided into two subcategories:

i) Explicit Feedback:- Packets are sent back from the point of Congestion to warn the source.

ii) Implicit Feedback:- The source deduces the existence of congestion by making local observations, such as the time needed for acknowledgements to come back.

## Congestion Prevention Policies :

→ Controlling congestion using open loop systems, where in systems are designed to minimize congestion in the first place, rather than letting it happen and reacting to it.

→ This can be achieved using appropriate policies at various levels

| Layer | Policies |
|---|---|
| Transport | • Retransmission policy<br>• Out-of-order caching policy<br>• Acknowledgement policy<br>• Flow control policy<br>• Timeout determination |
| Network | • Virtual circuits Vs Datagram inside the subnet<br>• Packet queuing and service policy<br>• Packet discard policy<br>• Routing algorithm<br>• Packet lifetime management |
| Data link | • Retransmission policy<br>• Out-of-order caching policy<br>• Acknowledgement policy<br>• Flow control policy |

Fig:- Policies that affect congestion

→ Data link layer :-

⇒ "Selective repeat" is better than "go back n" to control congestion at Data link layer for retransmission and buffering policy.

⇒ If each packet is ack immediately, then ack packets generates extra traffic. If "piggybacking" is used, then extra timeouts and retransmission may result.

⇒ A tight flow control scheme (eg, a small window) reduces the data rate and thus helps fight congestion.

→ Network Layer :-

⇒ Congestion control algorithm works with only virtual circuit subnets.

⇒ Routers have one queue per input line, one queue per output line or both. The order in which packets are processed (eg, Round robin / priority based).

⇒ Discard policy is the rule telling which packet is dropped when there is no space.

⇒ A good routing algorithm can help avoid congestion by spreading the traffic over all lines.

⇒ packet lifetime management deals with how long a packet may live before being discarded.

i) If it's too long → lost packets may clog up the work for a long time

ii) If it's too short → packets may time out before reaching destination.

→ Transport Layer :-

→ has similar issues as in data link layer.

→ ~~Determining~~ Determining timeout interval is difficult, because determining transit time across network is less predicatable than the transit time over a wire between two routers.

i) If time out is short → Extra packets will be sent unnecessarily

ii) If it is too long → Response time will suffer whenever a packet is lost.

## Congestion Control in Virtual Circuit Subnets :-

→ "Admission Control" technique is used to keep congestion that has already started from getting worse.

→ The idea is simple, once congestion has been signaled, no more virtual circuits are set up until the problem has gone away.

→ An alternative approach is to allow new virtual-circuits but carefully route all new virtual circuits around problem areas.

→ Another strategy relating to virtual circuits is to negotiate an agreement between the host and subnet when a virtual circuit is set up.

⇒ The agreement normally specifies, Volume and shape of traffic, QoS required, and other parameters.

⇒ As part of agreement, subnet typically reserve resources along the path when virtual circuit is set up.

⇒ Resources includes table and buffer space in routers and bandwidth on the lines.

⇒ Price of controlling congestion may result in wastage

## Congestion Control in Datagram Subnets

→ Each router can monitor the utilization of its output lines and other resources. It can associate with each line a real variable u, such that

$$u_{new} = a\,u_{old} + (1-a)\,f$$

Where,

u = reflects the recent utilization of that line lies between 0.0 to 1.0.

a = Constant that determines how fast the router forgets history.

f = A sample of instantaneous line utilization either 0 or 1.

Whenever u moves above threshold, the output line enters a "warning" state.

→ The Warning Bit :-

Each newly-arriving packet is checked to see if its op. line is in warning state. The following alternatives can be taken

⇒ Warning state is signaled by setting a special bit in the packet's header.

⇒ When the packet arrived at its destination, the warning bit is set into the next acknowledgement sent back to source. The source then cut back on traffic.

⇒ This process continues as long as the router is in the warning state. Source monitors the fraction of ack with the bit set and adjust its transmission rate accordingly.

Choke Packets :-

→ The router sends a choke packet back to the source host, giving it the destination found in the packet.

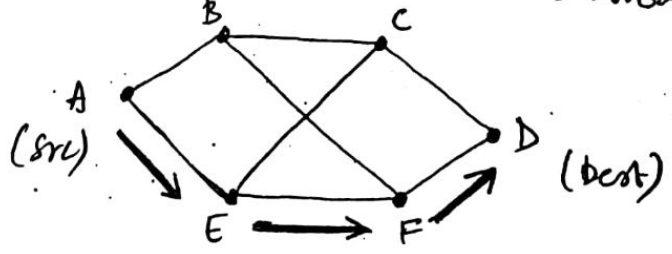
→ When the source host gets the choke packet, it is required to reduce the traffic sent to the specified destination by X percent.

→ After a fixed time interval, if the host receives a choke packet, then the line is still congested, so the host reduces the flow still more.

→ If no choke packet is received by host, then it may increase the flow again in small increments to prevent congestion from reoccurring quickly.

⇒ Hop-by-Hop Choke Packets :-

→ At high speeds or over long distances, sending a choke packet to the source hosts does not work well because the reaction is slow.




→ Consider a host in router A sends traffic to a host in router D at 155 Mbps.

→ If host at router D runs out of buffer, and say it will take 30 msecs for a choke packet to get back to host at router A. to tell it to slow down.

→ Even if host at router A completely shutdown immediately, the 4.6 megabits in the pipe will continue to pour in and have to be dealt with.

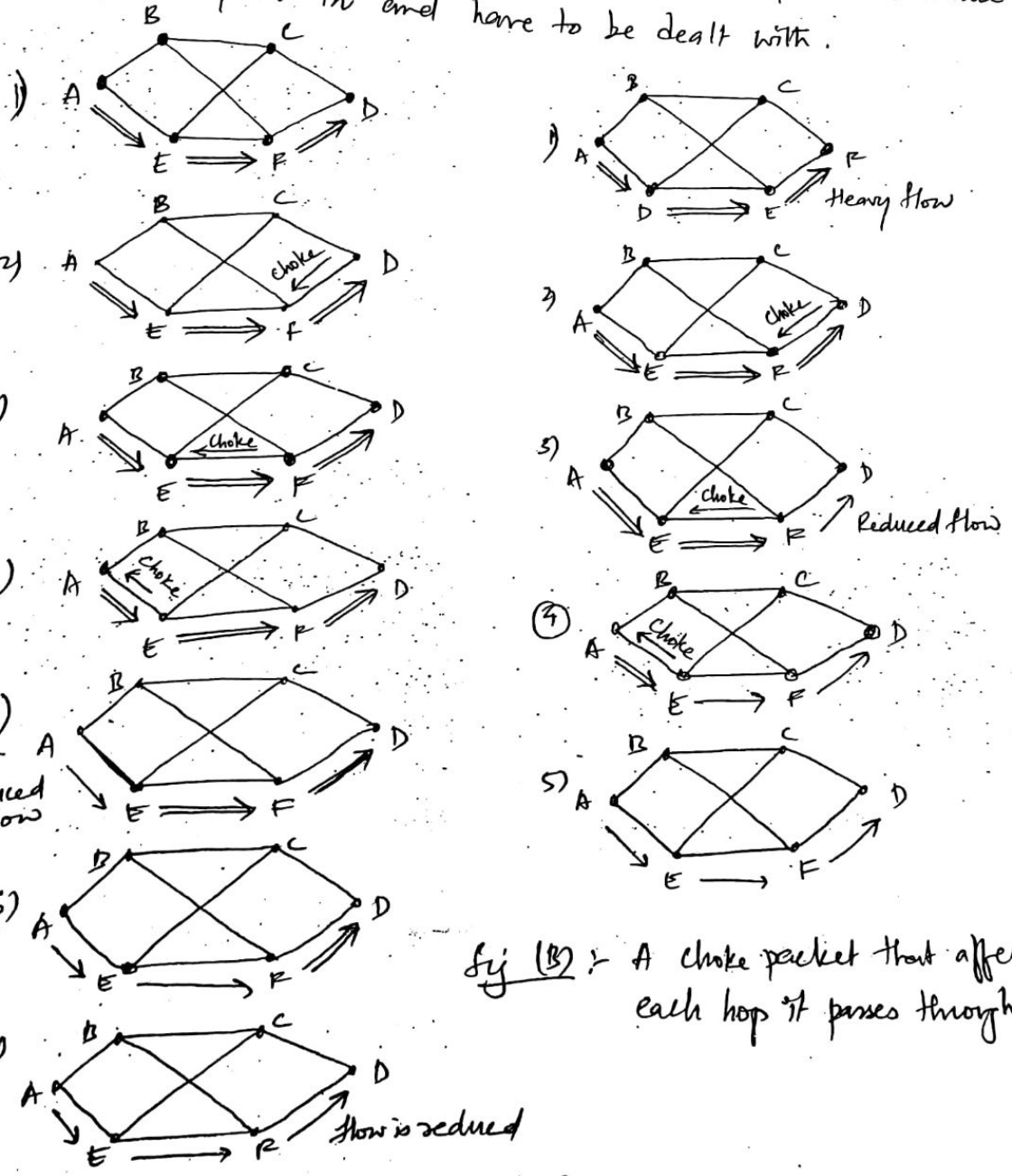



Fig (A):- Choke packet that affect only the Source

Fig (B) :- A choke packet that affects each hop it passes through

→ An Alternative is to have the choke packet effect at every hop it passes through.

→ Load Shedding :-

⇒ When routers are inundated by packets that they cannot handle, they just throw them away called Load Shedding.

⇒ Instead of dropping packets randomly, decisions can be taken based on application running

i) In the case of File transfer, keeping older packets and dropping newer packets results in less retransmissions.

ii) In the case of multimedia transfer, the new packet is more important than the old one

⇒ An intelligent discard policy can be developed with the help of sender telling him to mark a priority class to each packet.

⇒ Senders can be allowed to send high-priority packets under conditions of light load, as the load increases more they would also be discarded encouraging user to stop sending them.

## Random Early Detection (RED):-

⇒ Wired networks are very reliable, so lost packets are mostly due to buffer overflows rather than transmission errors.

⇒ This observation leads to the idea of discarding packets before all the buffer space is really exhausted (or buffer space threshold exceeded).

⇒ Routers can explicitly send choke packets to inform source about congestion. But doing so, it puts even more load on the already congested network.

Another strategy ⇒ Routers can discard the selected packets and doesn't inform about it to source. The source eventually notices lack of acknowledgement and deduces the existence of congestion in the network there by slowing down their transmission rate.

## Jitter Control:-

→ The variation (i.e., standard deviation) in the packet arrival times is called 'Jitter'.

→

→ For audio/video streaming applications, higher jitter value will give an uneven quality to the sound or movie.

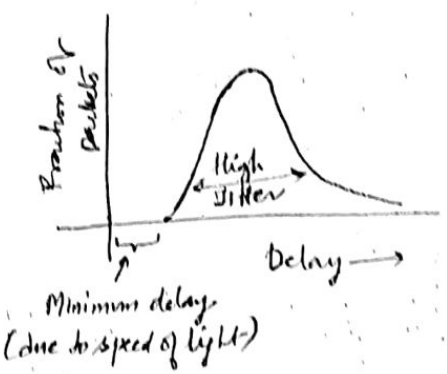



fig (a) : High Jitter

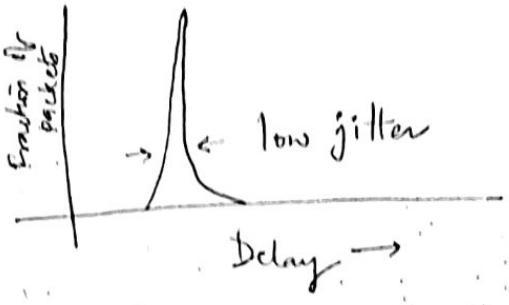



fig (b) : Low Jitter

→ Jitter can be bounded by computing expected transit time for each hop along the path. This information is stored in the packet and updated at each hop.

i) If packet is ahead of its schedule, router holds it just long enough to get back on schedule.

ii) If packet is behind the schedule, the router sends it quickly on the output line.

→ In some application, jitter can be eliminated by buffering at the receiver and then fetching data for display from the buffer. However, buffering is not applicable to real-time interaction such as video conferencing or Internet telephonic conversation.

# Quality of Service

→ With the growth of multimedia networking, serious attempts at guaranteeing QoS through network and protocol design are needed.

→
1. Requirements
2. Techniques for achieving Good QoS
3. Integrated Services
4. Differentiated Services
5. Label Switching and MPLS (MultiProtocol Label Switching)

→ **Requirements**:- {Flow:- A stream of packets from a source to a destination.}

⇒ QoS required by a flow can be characterized by four primary parameters:

i) Reliability
ii) Delay
iii) Jitter
iv) Bandwidth

| Application | Reliability | Delay | Jitter | Bandwidth |
|---|---|---|---|---|
| E-mail | High | Low | Low | Low |
| File Transfer | High | Low | Low | Medium |
| Web access | High | Medium | Low | Medium |
| Remote login | High | Medium | Medium | Low |
| Audio on Demand | Low | Low | High | Medium |
| Video on Demand | Low | Low | High | High |
| Telephony | Low | High | High | Low |
| Video conferencing | Low | High | High | High |
| | | | | |

Fig:- Stringent QoS requirements

→ ATM networks classify flows in four broad categories with respect to their QoS demands

i) Constant bit rate (eg Telephony)

ii) Real-time variable bit rate (eg., Compressed videoconferencing)

iii) Non-real-time variable bit rate (eg, watching movies over the Internet)

iv) Available bit rate (eg, file transfer)

→ Constant bit rate attempts to provide a uniform bandwidth and a uniform delay.

→ Variable bit rate occurs when video is compressed, some frames compressing more than the others

→ Available bit rate is for applications that are not sensitive to delay or jitter such as e-mail.

→ Techniques for achieving Good QoS :-

→ 1) Overprovisioning :-

⇒ An easy solution is to provide so much router capacity, buffer space and bandwidth that packets just fly through easily.

→ Disadvantage is the solution is expensive.

8) Buffering :-

⇒ Flow can be buffered on the receiving side before being delivered.

⇒ Buffering them does not affect the reliability or bandwidth and increases the delay, but it smooths out the jitter.

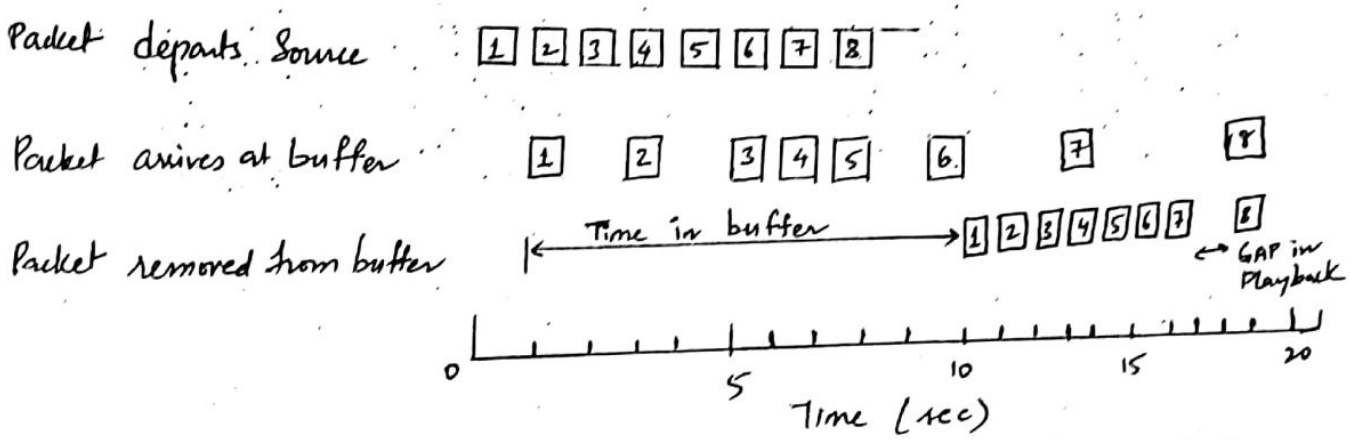



fig:- Smoothing the output stream by buffering packets

Traffic Shaping :-

⇒ Traffic shaping is about regulating the average rate of data transmission

⇒ When a connection is set up, the user and the subnet agree on a certain traffic pattern for that circuit called a "Service Level Agreement".

⇒ Monitoring a traffic flow from user is called "traffic policing", which can be done easily in Virtual-circuits subnet than Compare to Datagram Subnets.

⇒ As long as the user fulfills her part to the agreed-on contract, the subnet promises to deliver packets in a timely fashion.

⇒ Thus traffic shaping reduces congestion and helps the subnet to live up to its promises

→ Leaky Bucket Algorithm :-

⇒ Each host is connected to the network by an interface containing a finite internal queue (i.e, A Leaky bucket).

⇒ If one or more processes within the host try to send a packet when maximum number is already queued, the new packet is discarded. This can be built into the interface or simulated by the host operating system called the "Leaky bucket algorithm".

⇒ Leaky bucket algorithm turns an uneven flow of packets from the user processes inside the host into an even flow of packets onto the network, smoothing out bursts and greatly reducing the chances of congestion.

⇒ When similar size packets (eg. ATM cells) arrives and if there is room on the queue it is appended to the queue, otherwise it is discarded. At every clock tick, one packet is transmitted (unless queue is empty)

⇒ When packets are variable-sized, it is better to allow a fixed number of bytes per tick rather than just one packet.

eg. If rule is 1024 bytes per tick, then

(a) 1 – 1024 byte can be admitted
(b) 2 – 512 byte packet
(c) 4 – 256-byte packet

If residual byte count is too low, the next packet must wait until the next tick.

⇒ The byte-counting leaky bucket is implemented as:

i) At each tick counter is initialized to n.

ii) If first packet on the queue has fewer bytes than the current value of the counter, it is transmitted, and the counter is decremented by that number of bytes

⇒ iii) Additional packets may also be sent, as long as the counter is high enough

iv) When counter drops below the length of the next packet on the queue, transmission stops until the next tick.

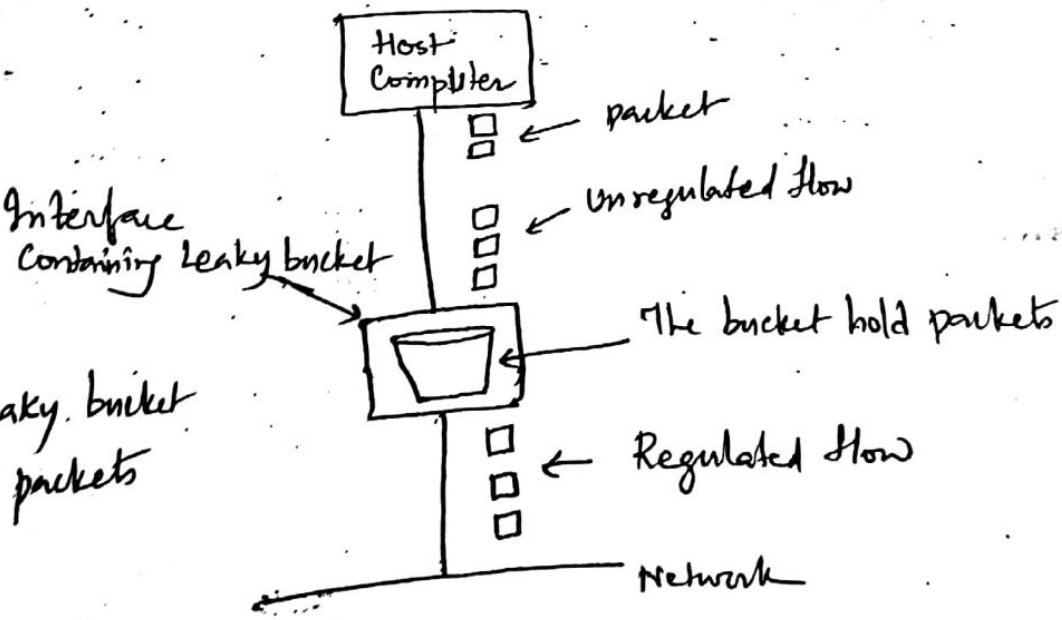
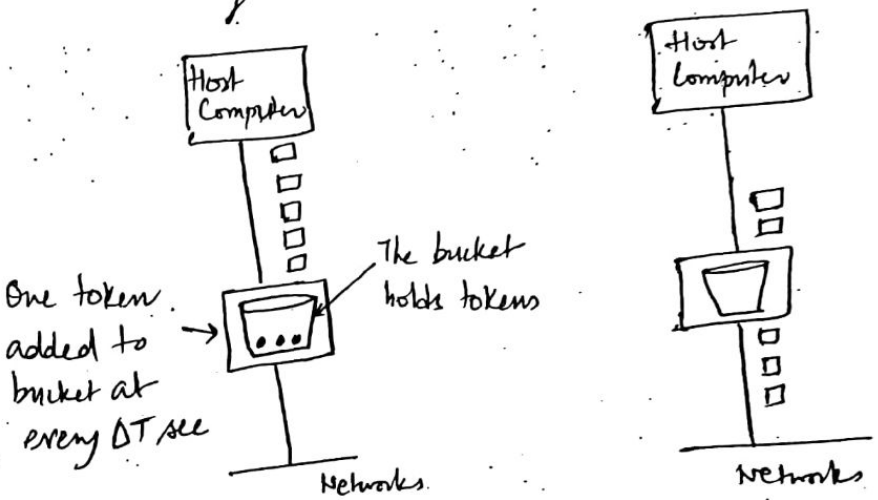



Fig:- A Leaky bucket with packets

## Token Bucket Algorithm :-

→ The leaky bucket hold tokens, generated by a clock at the rate of one token every $\Delta T$ sec.

→ For a packet to be transmitted, it must capture and destroy one token.




(a) Before (b) After

fig:- Token Bucket Algorithm

→ The above diagram demonstrates that three out of five packets got tokens and they got transmitted, while the other two stuck waiting for two more token to be generated.

→ Implementation requires just a variable that counts tokens.

i) Counter is incremented by 1 every $\Delta T$ sec

ii) Decremented by 1 whenever a packet is sent

iii) When counter hits zero, no packets may be sent

| Leaky Bucket Algorithm | Token Bucket Algorithm |
|---|---|
| 1) Discards packets when the bucket fills up. | 1) Discard tokens when bucket fills up but never discards packets |
| 2) Always sends packets at constant rate. | 2) Allows idle host to save up tokens and later can send large bursts at once. |
| 3) Bucket is a finite queue and outputs at finite rate | 3) If there is no token in bucket, packet cannot be send. |

→ One way to get a smoother traffic is to insert a leaky bucket after the token bucket. The rate of the leaky bucket should be higher than the token bucket but lower than the maximum rate of the network.

→ Resource Reservation :-

⇒ Once the shape of the traffic is regulated and a specific route for a flow is decided then it becomes possible to reserve resources along that route to make sure the needed capacity is available

⇒ Three different kinds of resources can potentially be reserved :

1) Bandwidth
2) Buffer space
3) CPU cycles

⇒ Bandwidth :- Reserving bandwidth means not oversubscribing any output line.

eg: A flow requires 1 Mbps & outgoing line has a capacity of 2 Mbps, trying to direct 3 flows through that line is not going to work.

⇒ Buffer space :- For good QoS, some buffers can be reserved for a specific flow so that flow does not have to compete for buffers with other flows.

⇒ CPU cycles :- It takes router CPU time to process a packet, so a router can process only a certain number of packets per second. Making sure that the CPU is not overloaded is needed to ensure timely processing of each packet.

## Admission Control :-

⇒ Many parties may be involved in the flow negotiation (the sender, the receiver and all the routers along the path between them), flows must be described accurately in terms of specific parameters called "Flow Specification".

⇒ A typical flow specification contains five parameters :

| Parameter | Unit |
|---|---|
| Token bucket rate | Bytes/sec |
| Token bucket size | Bytes |
| Peak data rate | Bytes/sec |
| minimum packet size | Bytes |
| Maximum packet size | Bytes |

fig :- An example flow specification.

⇒ (i) Sender produces a flow specification proposing the parameters it would like to use.

ii) Specification propagates along the route

iii) Each router examines it and modifies the parameter as needed. Modification can only reduce the flow, not increase it.

iv) When it gets to receiver, the parameters can be established.

⇒ Token bucket rate is the no of bytes per second that one put into the bucket

⇒ Token bucket size specifies bucket size and once tokens sent after that are lost

⇒ Peak data rate is the maximum tolerated transmission rate that sender must never exceed.

⇒ Minimum and maximum packet size including transport and network layer headers (eg. TCP & IP).

Min^m size is important as processing each packet takes fixed time, Max^m size is important due to internal network limitations that may not be exceeded.

⇒ Proportional Routing :-

⇒ In order to provide high QoS, The traffic for each destination can be splitted over multiple paths.

⇒ Since routers do not have overview of network-wide traffic, a feasible solution is to divide the traffic equally or in proportion to the capacity of the outgoing links.

## Packet Scheduling :-

If a router is handling multiple flows, then

i) One flow may occupy too much of its capacity and other flows may starve

ii) Aggressive sender may capture most of the router's queue

} Reduces QoS for others. Thus packet-scheduling algorithms have been devised

⇒ Fair Queuing algorithm :-

i) Routers have separate queues for each output line, one for each flow.

ii) When line becomes idle, router scans the queues round-robin, taking the first packet on the next queue.

Drawback :-

i) Algorithm gives more bandwidth to hosts that use large packets than to hosts that use small packets.

⇒ Above drawback is overcome by simulating a byte-by-byte round robin, instead of a packet-by-packet round robin.

⇒ Byte-by-Byte round robin gives all hosts the same priority, but it is desirable to give video servers more bandwidth than regular file servers, thus weighted fair queueing

→ Integrated Services :-

⇒ Generic name for devising an architecture for streaming multimedia is "Flow-based algorithms or Integrated services".

⇒ It is aimed at both unicast and multicast applications

⇒ An example of multicast application is a collection of digital television stations broadcasting their programs as streams of IP packets to many receivers at various locations.

⇒ ⇒ Groups can change membership dynamically, thus senders reserving bandwidth in advance does not work well as it require each sender to track all entries and exits of its audience (million subscribers).

⇒ RSVP :- The Resource reSerVation Protocol

⇒ RSVP is the main IETF protocol for integrated service architecture.

⇒ RSVP protocol is used for reservations ; other protocols are used for sending the data.

→ RSVP allows,

i) Multiple senders to transmit to multiple groups of receivers.

ii) Permits individual receivers to switch channels freely.

iii) Optimizes bandwidth use while eliminating congestion

The protocol uses multicast routing using spanning trees.

1) Each group is assigned a group address. Sender puts group's address in its packet to send to a group.
2) Standard multicast routing algorithm then builds a spanning tree covering all members.
(Routing algorithm is not part of RSVP).

⇒ To get better reception and eliminate congestion.

① Receivers in a group can send a reservation message up the tree to the sender.

② The message is propagated using the Reverse path forwarding algorithm

③ At each hop, the router notes the reservation and reserves the necessary bandwidth. By the time the message gets back to the source, bandwidth has been reserved all the way from sender to receiver making the reservation request along the spanning tree.

④ If insufficient bandwidth is available, it reports back failure.

⇒ If the host request to a random connect to a node already has a feed from an intermediate node then it does not have to reserve any more.

⇒ When does receivers share on path then the capacity reserved must be large enough to satisfy the greediest receiver.

## Differentiated Services:

- Drawback of "integrated services" is that they require an advance setup to establish each flow that does not scale well when there are thousands or millions of flows.
- ⇒ Routers also maintains an internal per-flow state making them vulnerable to crashes.
- ⇒ "Differentiated service" is an approach to provide QOS (which can be) largely implemented locally in each router without advance setup and without having the whole path involved. Also known as "class-based QOS".
- ⇒ Differentiated services (DS) can be offered by a set of routers forming an administrative domain (eg., ISP or telco).
- ⇒ Administration defines a set of services classes with corresponding forwarding rules.
- ⇒ Customer packets entering domain may carry a "Type of service" field in them, with better services provided to some classes (eg., premium service) than to others.
- ⇒ For packets, the classes may differ in terms of delay, jitter and probability of being discarded in congestion.

⇒ Expedited Forwarding :-

⇒ Defines a network-independent service class (since services are run by different operators).

⇒ Provides two classes of service :-
  i) regular
  ii) expedited

⇒ Vast majority of traffic is expected to be regular, but a small fraction of packets are expedited.

⇒ Expedited packets should be able to transit the subnet as if no other packets were present.

⇒ One way to implement this strategy is to program the routers to have two output queues for each outgoing line,
  i) One for expedited packets.
  ii) One for regular packets

Assured Forwarding :-

⇒ A scheme for managing the service classes.

⇒ It specifies that there shall be four priority classes, each class having its own resources and it defines three discard probabilities that are undergoing congestion : low, medium and high.

packet processing under assured forwarding :-

① Classify the packets into one of the four priority classes.

② Mark the packets according to their class. An 8-bit Type of service field available in IP Header (6-bits are are used for service class).

③ Pass the packets through a shaper/dropper filter that may delay or drop some of them.

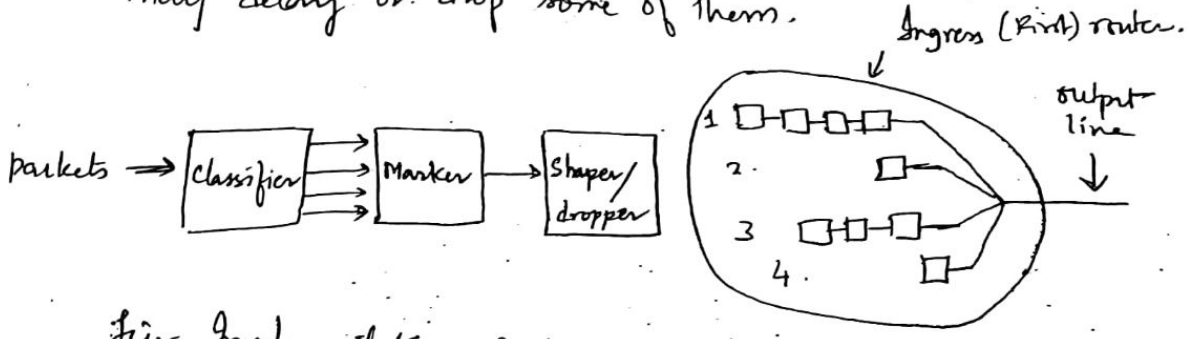



fig:- Implementation of the data flow for assured forwarding

⇒ Label Switching and MPLS :-

⇒ Adds a ~~label~~ label in front of each packet and doing the routing based on the label rather than on the destination address

⇒