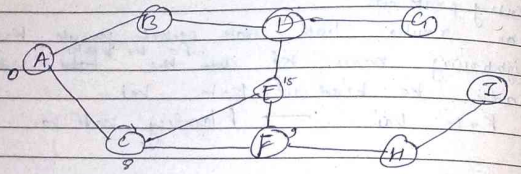
Link state routing has 2 phase

① Reliable flooding: initial state
↳ each node knows the cost of its neighbours.

② Route calculation,
↳ best route first known, using dijkstras algorithm.

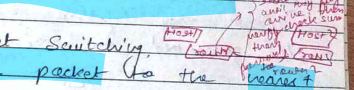


Hierarchical Routing
↳ It is a 2-level hierarchy.
↳ routers divided into regions.

UNIT-3

Network layer: Design issues, Routing algorithms: Dijkstra's algorithm, Bellman-ford algorithm, Link state Routing, Hierarchical Routing, Congestion Control Algorithms: General Principles of Congestion control, Prevention Policies, Congestion control in Virtual-Circuit Subnets, Congestion Control in Datagram subnets. QOS: technologies for achieving good QOS, Traffic Management, Integrated & Differentiated Services. RSVP.

Design Issues
→ Store & forward Packet Switching
→ Services provided to the Transport layer
→ Implementation of Connectionless Service
→ Implementation of connection Oriented Service

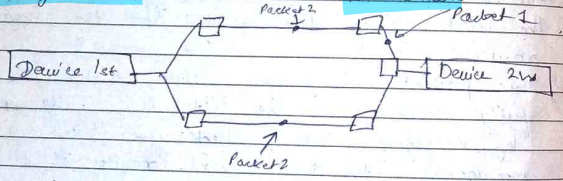


① Store & forward Packet Switching
- The host sends the packet to the nearest router
- This packet is stored there until it has fully arrived and the link is fully processed by verifying the checksum then it forwarded to the next router till it reached the destination.
- This mechanism is called store & forward Packet switching.

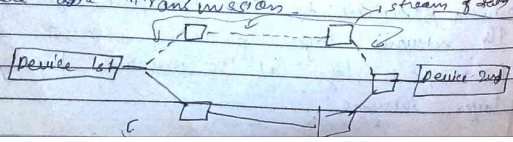
② Services provided to the transport layer.
- The network layer provides service to the transport layer at the network layer / transport layer interface.

→ The service need to be carefully designed with the goals in mind.
→ The service should be independent of the routers technology.
→ The transport layer should be shielded from the number, type & topology of the routers present.
→ The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs & WANs.

③ Implementation of Connectionless Service
→ No connection setup
→ Message is broken into packets & Each packet is individually routed
→ Routers decide line based on routing table.
→ Packets may follow different paths.
→ Not guaranteed to arrive in order.



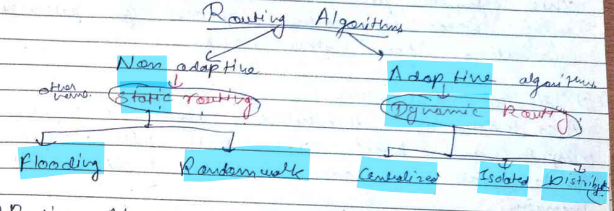
④ Connection oriented services
→ Connection is established beforehand.
→ Message is broken into packets & routed over the established route
→ Reliable data transmission



Routing Algorithms

Routing → Helps to decide which line pe incoming packet transmit karna chahiye.

→ Properties
→ Correctness, simplicity, stability, robustness, fairness & optimality.



⇒ Routing Algorithm is a procedure that lays down the route or path to transfer data packets from source to destination.
↳ The routing protocol is a routing algorithm that provides the best path from source to destination.
↳ Routing is the process of forwarding the packets from source to destination but the best route to send the packets is determined by the routing algorithms.

Adaptive Algorithms
↳ Known as dynamic routing algorithms.
↳ This algorithms makes the routing decisions based on topology & network traffic.
↳ The main parameters related to this algorithms are hop count, distance & estimated transmit time.

① Centralized algorithm.
It finds the least cost path b/w source & destination node by using global knowledge about the network. So it is also known as global routing algorithm.

② Isolated algorithm.
This algorithm process the routing information by using local information instead of gathering information from other nodes.

③ Distributed algorithm
Also known as decentralized algorithms as it computes the least - cost path b/w source & destination in an iterative & distributed manner.

Non - Adaptive Routing Algorithm.
- Also known as Static Routing Algorithm
- When booting up the network, the routing information stores to the routers.
- Non adaptive routing algorithms do not take the routing decisions based on the network topology or network traffic.

① Flooding :-
In flooding, when a data packet arrives at a router, it is sent to all the outgoing links except the one it has arrived on. Flooding may be uncontrolled, controlled or selective flooding.

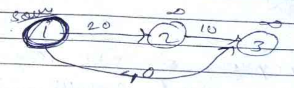
② Random walks :-
This is a probabilistic algorithms where a data packet is sent by the routes to any one of its neighbours randomly.

★ Dijkstra's Algorithm. (Single Source shortest path)
↳ used in Google maps.

* Relaxation.

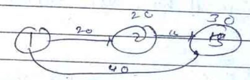
if $d(u) + c(u,v) < d(v)$
$d(v) = d(u) + c(u,v)$

Ex:-

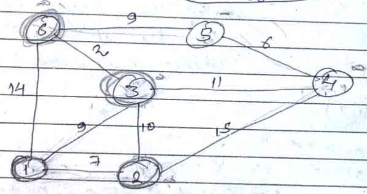


$d(u) + c(u,v) < d(v)$
$0 + 20 < \infty$
} Now ① → ② me ② ∞ se 20 ho gya.

Har path ko compare kro & shortest wala select kro



Q



source is ①

Source	Destination					
1	2	3.	4	5	6	
	∞	∞	∞	∞	∞	
	7 min fix.	9.	∞	∞	14	(1 से सबकी distance निकाली से minimum select krna
1, 2	7	9 min (9+17 में 9 ①→③→9 ①→②→③→17 small hai ① choose this)	22 (1→3→4)	∞	14 (1→3→6)	Ab 2 bhi source में आ गया ∴ 1, 2 will be fix now.
1, 2, 3	7	9	20	∞	11 min	
1, 2, 3, 6	7	9	20 min	20 (1→3→6→5)	11	Ab unse sabka distance calculate k
1, 2, 3, 6, 4/5	7	9	20			Dono same Koi bhi choose kr lo / min lelo

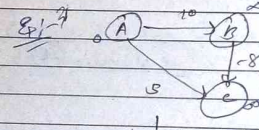
∴ Final sequence is - 1, 2, 3, 6, 4, 5.
(1, 2, 3, 6, 5, 4)

Time complexity = $O(V) + O(V) + V \log V + E \log V$ (biggest is)
$= O(E \log V)$

Failure / Disadvantage
↳ Not work properly on -ve edges.
It may give correct answer or may not give.

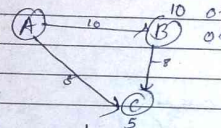
(2) Bellman-Ford Algorithm
Har edge ke (V-1) no. of time relax krna hai.

Eg:



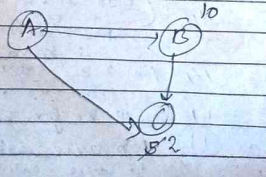
starting me source 0 & destinations ∞ distance pe hai.

R1 (relax 1)



0+10 < ∞ yes ∴ B is now 10.
0+5 < ∞ yes ∴ C is now 5
Now B to C ∞ -8 is ∞ ∴ no change
(Previous wala Diagram ko dekh ke relax kro)

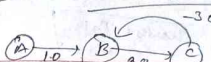
R2



2 times me relax kr ke de diya.

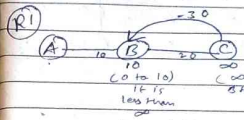
(Ek aur baar relax kr ke bhi same hi answer aaya)

Question 2 solve by Bellman-Ford.



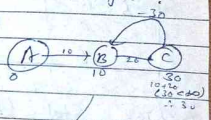
(V-1) ⇒ (3-1) = 2 times relax kr diya.

R1



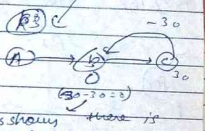
(0 to 10) if is less than ∞ ∴ assign 0
(∞+20) B to C previous ko dekh ke.

R2



(10-30 < 20) ∴ 3rd

R3

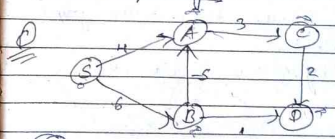


(30-30 = 0)

(V-1) times check & +1 times cycle to check for -ve cycle.
(V-1) + 1

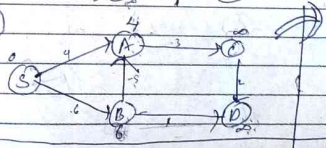
This shows there is no -ve edge cycle ∴ we will not get min ans wer.

Time complexity = O(V.E)

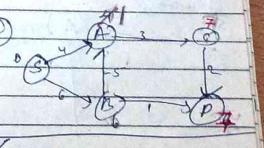


Total no. of relaxations
= |V-1|
= |5-1| = 4

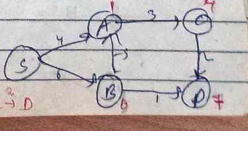
R1



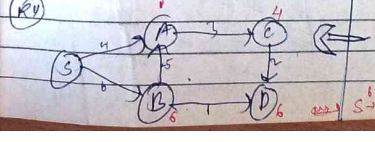
R2



R3

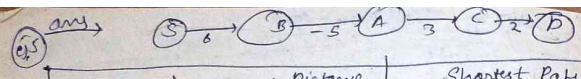


R4



S → B → A → C → D

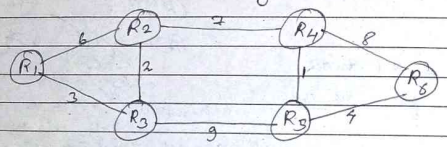
ans →



Path	Shortest Distance	Shortest Path
S-A	1	S-B-A (6-5=1)
S-B	6	S-B (6)
S-C	4	S-B-A-C (6-5+3)
S-D	5	S-B-A-C-D (6-5+3+1)

Date: / / Page no:

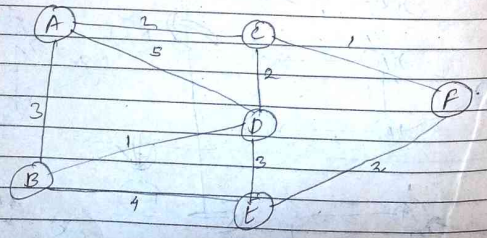
Link State Routing



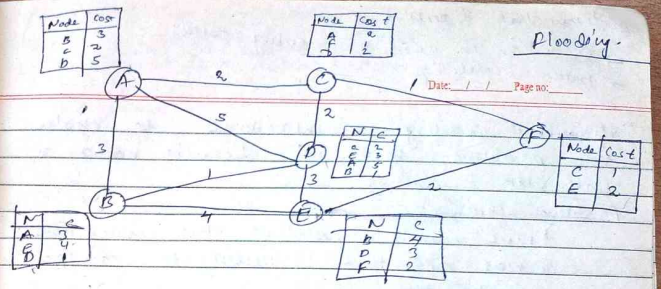
2 phases ① Reliable Flooding.
- ↳ Each node knows the cost of its neighbours
- ↳ Each node knows the entire graph

② Route Calculation
- ↳ using Dijkstra's algorithm.

Question:

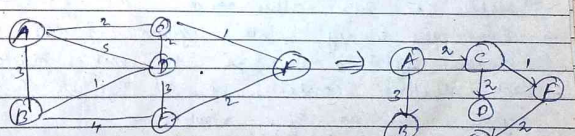


Flooding.



Now, Dijkstra's Algorithm

Source	Destination's				
	B	C	D	E	F
A	3	2	5	∞	∞
A, C	3	2	4	∞	3
A, C, B	3	2	4	7	3
A, C, B, F	3	2	4	5	3
A C B F D	3	2	4	5	3
A C B F D E	3	2	4	5	3



path.	Distance	via.
A → B	3	A-B
A → C	2	A-C
A → D	4	A-C-D
A → E	5	A-C-F-E
A → F	3	A-C-F

Hierarchical Routing -

→ Reduction in size of routing tables.
→ better scalability

→ Hierarchical routing is introduced to resolve the problems faced in distance vector or link state routing.

Problems faced are:-
1) more memory needed to store routing tables
2) more CPU time is needed to scan each routing table
3) more bandwidth required to send scanning reports

In hierarchical routing, routers are classified in group known as region.
→ Each router has only the information about the routers in its own region & has no information about routers in other regions.

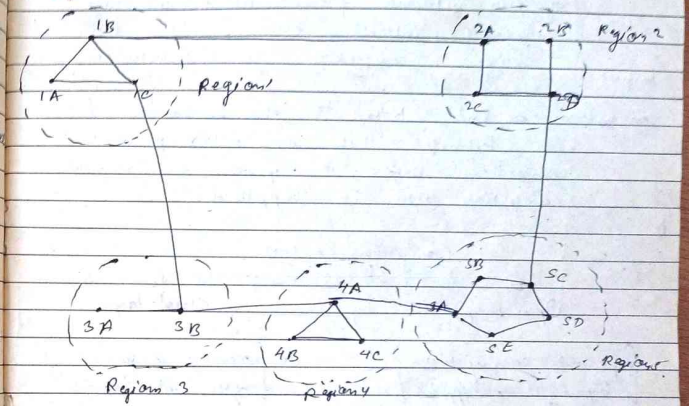
routers → group of → regions → clusters (if regions are big)
(combination of nodes or regions)

→ Routers just save one record in their table for every other region.
→ Even after dividing into regions if the network is too big then we go to make clusters.
→ Each cluster contain no. of regions & each region contains no. of routers.

Hop → transfer from 1 node to another node.

1K router apne ki region ke router ko share kr skta h but they are totally unaware of how to route packets to regions routers of other regions.

To overcome this problem we use Gateway router. Gateway router contain information about other regions router as well.



Full table for 1A

Destination	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	3
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	5
5B	1C	5
5C	1B	4
5D	1C	5

Hierarchical table for 1A

Destination	Line	Hops
1A	—	—
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Congestion Control Algorithms

Congestion is a network may occur if the load of the network is greater than that capacity of the network.

Ex:- Doubledig capacity is more than capacity of our network from which we are downloading, internet become slow then this is congestion in the network.

Network slow → delay → data loss may be due to traffic.

Congestion control:- Refers to the mechanism & techniques that can either prevent congestion before it happens or remove congestion after it happened.

Congestion Control

- Open Loop
  - prevent congestion before it happens
    - Retransmission policy
    - Window policy
    - Acknowledge policy
    - discarding policy
    - Admission policy
- Closed loop
  - remove congestion after it happens
    - back pressure
    - choke packet
    - implicit signaling
    - explicit signaling
    - Load shedding

Open Loop

① Retransmission → packet can be retransmitted if sender feels it is lost or corrupted.

② Window → we selective reject window method for congestion control.

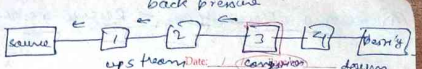
③ Acknowledgement Reciver → acknowledges → sender.

④ discarding policy → Routers discard less sensitive packets when congestion is likely to happen.

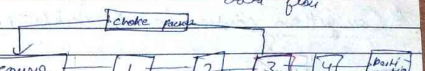
⑤ Admission policy :- Quality of service mechanisms.

Closed Loop-

① back pressure



② choke packet



③ Load shedding

Routers buffer becomes full, routers (discard packets)

Techniques for policing: Cold is better than no way → warm policy, milk (new is better than old) → milk policy

④ Implicit signaling

source guesses there is a congestion in network when it does not receive any acknowledgement.

Source slows down.

⑤ Explicit signaling → Forward signalling, Backward signalling

sending direct signal to source or destination (forward or backward direction).

monitor network traffic, detect congestion & take corrective action to prevent network collapse.

General principles of congestion control:-

→ The primary goal is to manage network traffic to prevent congestion & ensure efficient data transfer.

→ Principles include traffic shaping, load balancing, resource reservation & feedback mechanisms

↳ ensure fair resource allocation among users

↳ minimize packet loss & delay.

Prevention policies:-

① Traffic shaping:-

Techniques like leaky bucket & token bucket algorithms are used to control the rate of data transmission.

② Admission control.

Determining whether a new data flow can be accommodated without degrading service for existing flows.

③ Priority schemes → Assigns different levels of priority to different types of data.

(4) Resource Reservation → Reserve resource for priority traffic

Date: / / Page no:

☆ Congestion control in Virtual-Circuit subnets

	Datagram subnet	Virtual circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address.	Each packet contains a short VC number.
Routing	Each packet is routed independently.	Route chosen when VC is setup, all packets follow it.
QoS	Difficult.	Easy.
Congestion control	Difficult.	Easy.

☆ Congestion Management Policies.

(1) Traffic Shaping
- → Leaky Bucket
- → Token Bucket

Traffic shaping is a network (Congestion) management technique
- → Mechanism to control the amount & the rate of the traffic sent to the network.
- → Also known as packet shaping

→ Regulate rate of Data transmission.
(Helps to reduce congestion)
(Priority application are delivered without delay.

→ To increase network performance.

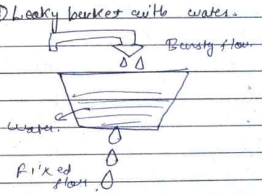
Leaky Bucket Algorithm.

It shapes bursty traffic into fixed rate traffic by averaging the data rate.

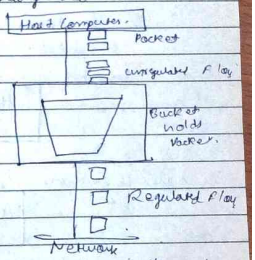
Date: / / Page no:

→ Bucket with a hole at bottom.
→ Flow of the water from bucket is at a constant rate which is independent of water entering the bucket.
→ If bucket is full, any additional water entering in the bucket is thrown out.
→ The input rate can vary, but the output rate remains constant.

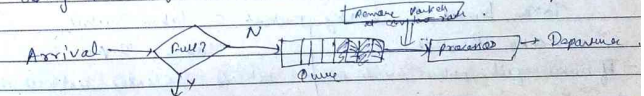
(a) Leaky bucket with water.



(b) Leaky bucket with Packet.



→ A simple leaky bucket algorithm can be implemented using FIFO queue. A FIFO queue holds the packets.



Problem with leaky bucket.
Leaky bucket algorithm allows only an average (constant.) rate of data flow. Its major problem is that it cannot deal with bursty data. To overcome this we have token bu

Token bucket Algorithm.

→ Allows bursty data transfer.

→ Modification of Leaky bucket / Overcomes leaky bucket

Contains tokens

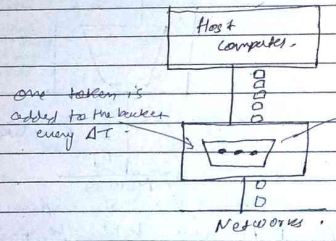
→ A token (s) are generated at every clock tick. For a packet to be transmitted, system must remove tokens from the bucket.

Steps: ① In regular intervals tokens are thrown into the bucket.

② The bucket has a maximum capacity.

③ If there is a ready packet, a token is removed from the bucket, & the packet is sent.

④ If there is no token in the bucket, the packet cannot be sent.



Token bucket + Leaky Bucket = Token Bucket

Leaky Bucket	Token Bucket
→ If bucket is full, packets are discarded	→ tokens discarded not packets
→ Token independent	→ Token dependent.
→ Does not save token	→ Saves tokens to send large burst
→ It has priority policy	→ Does not have
→ cannot deal with bursty data.	→ can deal

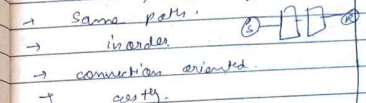
Open Loop - (Prevention)

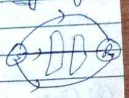
1) Retransmission

(2) Window policy: [1][2][3] → [2]

Only 2nd packet rendered so by using window policy it will know to send the specific packet that have been lost.

Virtual circuit	Datagram
→ Bandwidth, CPU, Buffer	→ No.
→ Same path.	→ out of order, May arrive not
→ in order	→ out of order
→ connection oriented	→ connectionless.
→ costly.	→ Not costly.
→ Reliable.	→ Not Reliable.





★ Congestion control in Virtual circuit:

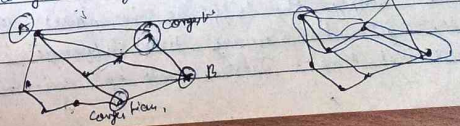
These are closed-loop based designed for virtual circuits subnets, which are connection oriented during connection set up, something can be done to help congestion control.

① Admission control:-

Once congestion has been signaled, no more new virtual circuits can be set up until the problem has gone away.

② Alternative Routes.

To avoid part of the network that is overloaded i.e. temporarily rebuild your view of network.



Congestion control in Datagram subnet

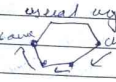
① The warning bit

When a new packet is to be transmitted on the output line router is at warning state, a special bit is added in header to signal this state. At the destination the information is sent back with ACK to the sender so that it could cut the traffic. When warning bit is absent, sender increases its transmitting rate.

Note: It uses a whole trip (source → destination → source) to tell the source to slow down.

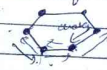
② Choke packet Technique.

The router sends choke packet back to the source host. The original packet is marked so that it would not generate any more choke further along the path & is then forwarded in the usual way.



③ Hop-by-Hop choke Packets

In this approach, unlike choke packet, reduction of flow start from intermediate node, rather than source node. The net effect of hop-by-hop scheme is to provide quick relief at the point of congestion.



④ Load Shedding

Router just throws packet away.

⑤ Jitter control:-

Jitter is simply the difference in packet delay. In other words, jitter is measuring time difference in packet inter-arrival time. Jitter should be minimized.

☆ Quality of Service (QoS)

The needs of each flow can be characterized by 4 primary parameters: bandwidth, delay, jitter & loss.

→ We try to create an appropriate environment for traffic.

↳ characterization

Reliability | delay | Jitter | Bandwidth

① Reliability :- Lack of reliability means losing a packet or ack (usually high) if it happens, the packet is retransmitted which declare reliability. Ex email, file transfer.

② Delay → source to destination delay.

Techniques for achieving good QoS

1) Traffic shaping
2) Resource Reservation
3) Packet Scheduling
4) Buffer management
5) Policing & Mapping

Traffic management :- Load balancing, Congestion control, Admission control.

QoS architecture model :-

☆ Integrated & Differentiated services :-

Integrated Services (IntServ)

→ Provides end-to-end QoS by reserving resources for each flow.

→ uses RSVP (Resource Reservation protocol) to signal resource requirements to ensure a guaranteed level of service.

→ used to establish dynamic RSVP over networks.

Int Serv

① RSVP → Resource Reservation Protocol.
↳ End-to-end QoS guarantees.

② C-BQ → Class-Based Queuing.
↳ Prioritize traffic.

③ Real-Time Transport Protocol (RTP)
↳ Ensure timely delivery.

★ Differentiated Service (Diff serv).

① Classify traffic → priority classes.
② Assign DSCP → Differentiated treatment.
③ Queue management → Prioritize traffic.
④ Traffic conditioning → Enforce policies.

Key Protocol:- ① RSVP, ② MPLS ③ Diff Serv
④ ECN ⑤ RED/WRED

★ RSVP:-
Resource reservation protocol is a network protocol that helps ensure that the network can provide the necessary bandwidth & performance for applications.
RSVP is used to signal the QoS needs of application traffic & to reserve resource across a network.

Features:-
① Receiver-initialized.
② Simplex flows
③ QoS.