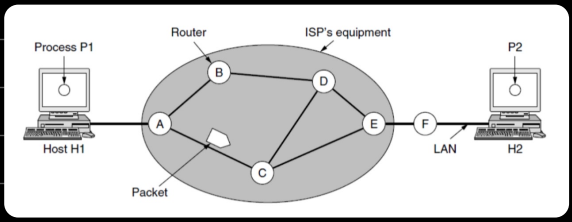
Unit-II : Network layer design Issues:

⇒ network layer is concerned with getting the packets from the source all the way to the dest.
⇒ it is the lowest layer that deals with end to end transmission.
⇒ it provides the best path, and must know the entire topology of the network. i.e the set of routers, links etc.
⇒ it has to chose appropriate path for the packet and avoid overloading.
⇒ when source and dest are in diff networks, new problems occur.

Store and forward packet switching:



⇒ ISP's equipments are routers connected by link.
⇒ shaded oval contains ISP equipments.
⇒ Customer equipment (shown outside the oval).
⇒ These 2 equipments are imp for store & forward packet switching.
⇒ The host H1 is connected to one of the ISP routers 'A', say a home computer plugged to a DSL modem. Consider H2 as host2, is connected to a LAN which might be office ethernet with a router 'F' owned and operated by the customer. This router has a leased line to the ISP's equipment. 'F' is outside as it does not belong to ISP.
⇒ The host1 transmits the packet to the nearest router. The packet is stored there until it has fully arrived & processed by verifying the checksum (it is the value that rep no. of bits in a message).
⇒ Then, the packet is forwarded to the next router along the path until it reaches the destination host.
⇒ This mechanism is store and forward packet switching.

Services provided to the transport layer:

⇒ The network layer provides services to the transport layer.
⇒ These services need to be designed carefully with foll goals:
1. services should be independent of the router technology.
2. Transport layer should be shielded from the no., type, topology of the routers
3. the network add made available to the transport layer should use a uniform numbering plan.

⇒ there are two diff philosophies about the services of network layer:

① <u>The internet community's perspective:</u>

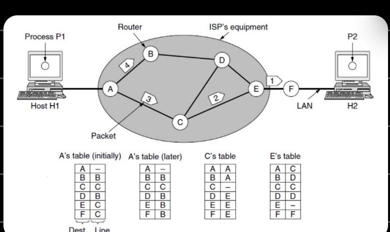
⇒ primary role of routers is to forward the packets to their dest as quickly & efficiently as possible.
⇒ routers must not be responsible for ensuring the reliability of transmission.
⇒ The hosts are resp for that (error, flow control). The hosts need to handle packet loss, duplication, data integrity, reordering.
⇒ This allows routers to be simpler and faster since they do not need to do additional tasks.
⇒ The network should be connectionless.
⇒ Each packet must carry the full dest add as each packet sent is carried independently.

② <u>The telephone company's perspective:</u>

⇒ the network should be connec oriented to ensure a reliable path b/w the source and the dest.
⇒ QoS ⇒ quality of service plays an imp role and without connections in network, the QoS is very difficult to achieve especially req for real-time data like voice and video.
⇒ flexibility is more imp than reliability.

Implementation of Conectionless service:

⇒ packets are routed independently of each other.
⇒ no advance setup is needed (as told above). In this method the packets are often called <u>DATAGRAMS</u>. The network is known as <u>datagram network</u>.
⇒ Suppose P1 has a long message for P2.
⇒ It hands the message to the transport layer with instructions to deliver it to P2 on Host 2.
⇒ If the message is very large, the network layer breaks it into 4 packets 1, 2, 3, 4 and sends each of them in turn to router 'A' using some P.P. Protocol. At this point the ISP takes over.
⇒ every router has an internal table telling it where to send packets for each of the possible dest.
⇒ each table has 'dest', 'outgoing line' to reach the dest. Only directly connected lines can be used.
ex: A has only 2 lines → B, C that can be used even if the end dest is some other router.
⇒ each packet is forwarded acc to the internal table.
⇒ it takes a diff route/path due to traffic, if route is crashed etc. This is done by an algorithm ⇒ ROUTING ALGORITHM ⇒ manages internal tables.
ex: IP is a connectionless network service.



<u>A's internal tables:</u>

	next router name	path taken
for A, no route is req for path.	A	—
To approach B from A, 'B' itself is path	B	B
To approach C from A, 'C' itself is path	C	C
To approach D from A, 'B' is path.	D	B
To approach E from A, 'C' is path.	E	C
To approach F from A, 'C' is path	F	C

initially

next router name	path taken
A	—
B	B
C	C
D	B
E	B
F	B

later (due to crash/traffic)

} if C is crashed/traffic etc

Implementation of Connec- oriented protocol:

⇒ in connec- oriented method, a path from source to dest has to be established b4 data can be sent. (virtual path)

⇒ This connec is called a VIRTUAL - CIRCUIT NETWORK.

⇒ virtual circuits are used to avoid having independent routes for every packet sent unlike what we saw above.

⇒ when a connec is estab, a route from source → dest is chosen & stored in tables inside routers.

⇒ this route is used for all traffic flowing over the connec. ex: telephone system working

⇒ when the connec is released, the virtual-circuit /network is also terminated.

⇒ each packet carries an IDENTFIER telling which virtual circuit it belongs to.

⇒ In fig, H1 establishes connec 1 with H2. This connec is remembered as the 1st entry in each of the tables in the router.

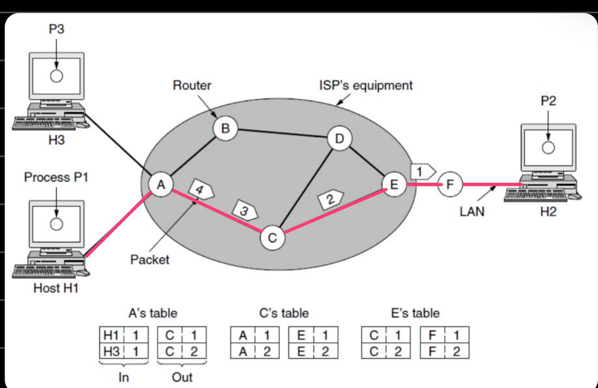
⇒ ROUTING TABLES maintain entries for each active connec.

⇒ virtual circuit is estab from H1 to H2. In A's table it says if a packet of identifier 1 comes from H1, it is to be sent to C & given identifier 1.

⇒ now if H3 also wants to estab connec with H2. It chooses identifier 1 as it is the only initial connec present). H3 tells to network to estab a virtual circuit, now, there are 2 rows in the tables. This leads to a conflict because even if A can distinguish connec 1 packets from H1 and connec 1 packet from H3, C cannot do it. So, A assigns a diff connec identifier to the traffic for the 2nd connec. This is known as Label Switching ?

"The ability of router to replace connec identifiers in outgoing packets"!

Ex: MPLS (multi protocol label switching) is a connec-oriented network. It used within ISP networks in the internet.



A's table:

in		out	
H1	1	C	1
H3	1	C	2

(i) packet comes from H1 with identifier 1 and it has to send it to C through identifier 1 only)

(ii) A receives packet from H3 with identifier 1, then A cannot send the packet to C with the same identifier, so, it changes it to identifier 2. this is known as LABEL SWITCHING)

Comparision of virtual-circuit & datagram-network:

datagram network	virtual - network
⇒ virtual path / circuit estab not needed.	⇒ virtual path / circuit estab from source to dest is needed.
⇒ each packet is routed independently to its dest.	⇒ all packets follow the route setup by the virtual network.
⇒ each packet has full source and dest address.	⇒ packet has only the VC number / short VC no.
⇒ QoS is difficult to guarantee	⇒ QoS is easier to guarantee.
⇒ packets may arrive out of order.	⇒ packets arrive in order.
⇒ highly flexible and reliable	⇒ less flexible & reliable.
⇒ simple routers, no identifiers used.	⇒ more complex routers, identifiers are used.
ex: IP	ex: MPLS.

Routing Algorithm:

⇒ main func of the Network layer is to route packets from source to dest.

⇒ Routing Alg is the network layer software responsible to decide which output line / outgoing link it should be transmitted on.

⇒ Properties of routing algs:

1. Correctness : determines the best / appropriate route.
2. Simplicity : easy to implement.
3. robustness : can handle changes in traffic without disruptions.
4. efficiency : min delays, max throughput.
5. fairness : ensure equal dist of resources.

⇒ In datagram-network, each packet's route is decided independently.
⇒ In virtual-network, routes are decided when a virtual circuit is estab. and remain fixed for a time period ⇒ session time and this is known as session routing. (a route remains active / in force for an entire session).

⇒ routing decides which route to use involving creation & updating of routing tables but the process of handling each incoming packet & determining the outgoing line using the routing table is called forwarding.

⇒ Routing algs can be grouped into 2 major classes:

1. Nonadaptive   2. adaptive.

⇒ Nonadaptive (static) algs: (simpler)
* do not take routing decisions based on current topology & traffic.
* routes are pre-computed in advance and downloaded to routers when the network is booted.
* does not adapt to failures/changes in the network.
* useful when the routing choice is clear/unlikely to change.

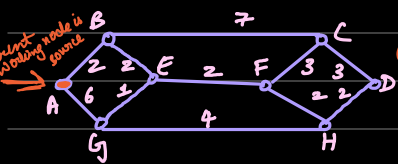
⇒ Adaptive (dynamic) routing alg: (complex)
* takes routing decisions based on changes in the topology & traffic.
* can update routes when topology changes or periodically based on the real time network conditions.
* uses metrics like distance, transit time for determining the best route.
* can get data from local routers, adj routers, all routers in the network.

Shortest Path Routing | Dijikstras Alg:

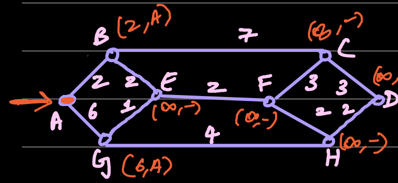
⇒ simple technique for computing optimal paths in a network.
⇒ these paths are the ones we want a distributed routing Alg to find.
⇒ The idea is to build a graph of network with each node of the graph representing a router and each edge of the graph rep a comm link / line.
⇒ To choose a route b/w a given pair of routers, the alg just finds the shortest path b/w them on the graph.
⇒ initially all paths are labelled as ∞ as in tentative, later they are made permanent.

⟹ find the shortest path from source router 'A' to dest router 'D'. Other nodes are tentative nodes.

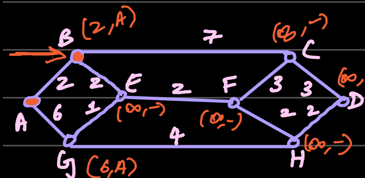
current working node is source



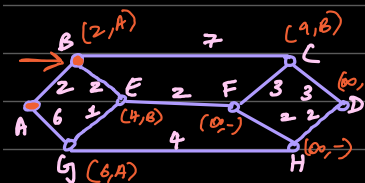
(make source node as permanent)



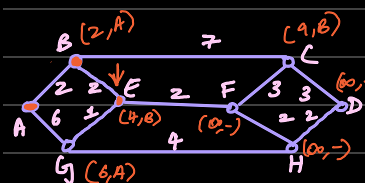
⟹ neighbours of 'A' are B and G.
⟹ note the cost of both.
⟹ mark the ones we dont know as (∞, −)



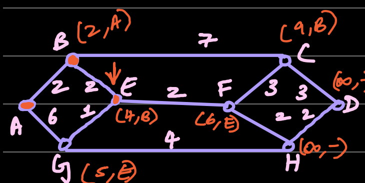
⟹ we know that '2' is smaller value out of 2 & 6.
⟹ make 'B' as permanent node/current working node.



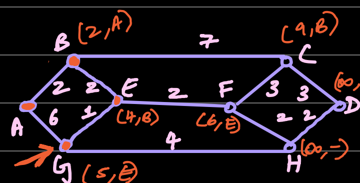
⟹ explore 'B'.
⟹ neighbours are 'E' and 'C' ⟹ note the costs.
⟹ make sure to add dist of the prev router too.



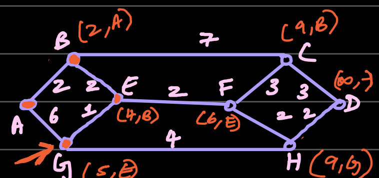
⟹ out of 4, 6 & 9, smallest is 4.
⟹ make 'E' permanent / current working node



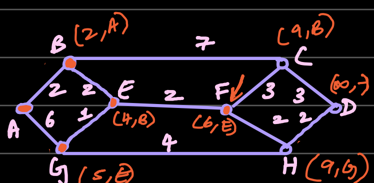
⟹ explore E ⟹ G and F.
⟹ G was already determined by 'A', change it wrt 'E'.



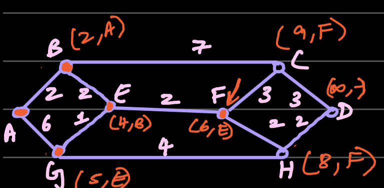
⟹ out of C, F and G; 'G' is smallest.
⟹ make 'G' current node.



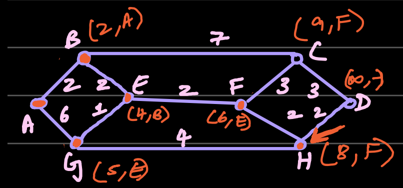
⟹ explore 'G' ⟹ 'H'.
⟹ mark cost of H.



⟹ out of H, C and F, 'F' is smallest.
⟹ make 'F' permanent.

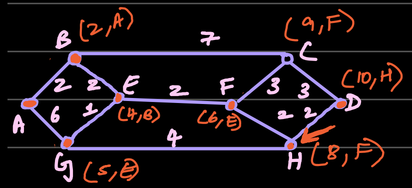


⟹ explore 'F' ⟹ C & H. (change values)
⟹ mark changed costs.



⇒ H is smallest.

⇒ make H ⇒ current node.



⇒ explore 'H' ⇒ D.

⇒ mark cost.

⇒ the path is D ← H ← F ← E ← B ← A //.

Use cases:

⇒ network routing.

⇒ GPS navigation systems.

⇒ healthcare resource allocation.

⇒ game development : AI pathfinding.

⇒ telecomm cable network design.

Flooding:

⇒ local routing technique.

⇒ every incoming packet is sent on every available outgoing line except the one it arrived on.

Challenges:

⇒ generates many duplicate packets.

⇒ So, we have to take measures.

⇒ One such measure is to have a hop-counter in the header of each packet. The counter is decremented at each hop the packet is discarded when the counter reaches zero. The counter should be initialized to the estimated path length (source → dest).

⇒ Still, duplicates cannot be avoided.

⇒ To further reduce duplicates, routers can maintain a list of sequence nos for each packet ensuring the source router doesn't forward packets they have already seen. The source router put a sequence no. in each packet it receives from its hosts.

⇒ Then, each router needs a list per source router telling which sequence numbers originating at that source have already been seen. If an incoming packet is on the list, it is not flooded.

Benefits:

⇒ guaranteed delivery ⇒ ensures packet reaches every node in network, better for broadcasting info, delivers packet to every node.

⇒ robustness ⇒ highly reliable, even when a router fails it will find a path.

⇒ minimal setup ⇒ routers only need knowledge of their immediate neighbours. Therefore, req min setup.

⇒ Shortest path guarantee ⇒ effectively finds shortest path by exploring all possible paths.

Use-Cases:

⇒ wireless networks ⇒ broadcasting messages ⇒ guaranteed delivery factor.

⇒ military / emergency networks ⇒ robustness factor.

⇒ to check efficiency of other routing algs ⇒ Shortest path guarantee factor //.

Distance - Vector Routing: (dynamic routing algorithms)

router = hop

⇒ aka distributed Bellman-ford Algorithm.
⇒ How does it work?

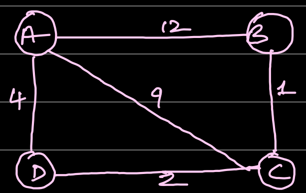
① Routing Tables:

* each router maintains a routing table with entries for every other router in network.
* each entry has two key info pieces:
  - i) next hop : outgoing link to reach the dest.
  - ii) cost : distance to that dest.
  - iii) dest router

② Dist Vector Alg:

* each router updates its own routing table choosing the best path (least cost).

ex:



find all the routing tables for all routers.

(note: 4 - 1 = 3 iterations)

Step ①

A:

dest router	dist	next router
A→A A	0	A
A→B B	12	B
A→C C	9	C
A→D D	4	D

(i) source & dest routers are same dist is 0)

B:

dest router	dist	next router
B→A A	12	A
B→B B	0	B
B→C C	1	C
B→D D	∞ no path	— no router

C:

dest router	dist	next router
A	9	A
B	1	B
C	0	C
D	2	D

D:

dest router	dist	next router
D→A A	4	A
D→B B	∞	—
D→C C	2	C
D→D D	0	D

Step ② each router will share its routing table info with its neighbours i.e only the middle column is shared.

A:

dest router	dist	next router
A	0	A
B	7	D, C
C	6	D
D	4	D

B has info of B, C, D info/dist vectors.
$A \to B = 12$
$\underset{9}{A \to C} + \underset{1}{C \to B} = 10$
$\underset{4}{A \to D} + \underset{2}{D \to C} + \underset{1}{C \to B} = 7$
} smallest is 7 so, take 7 as dist.

B:

dest router	dist	next router
B→A A	7	C, D
B→B B	0	B
B→C C	1	C
B→D D	3	C, D

A has B, D as neighbours.
$B \to A = 12$
$\underset{1}{B \to C} + \underset{9}{C \to A} = 10$
$\underset{1}{B \to C} + \underset{2}{C \to D} + \underset{4}{D \to A} = 7$
} smallest is 7

$B \to A + A \to D = 16$
$B \to C + C \to D = 3$
} smallest is 3

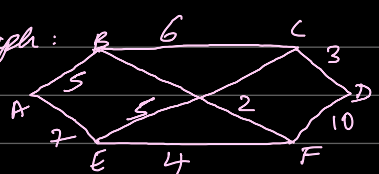
C:

dest router	dist	next router
A	6	D
B	1	B
C	0	C
D	2	D

D:

dest router	dist	next router
D→A A	4	A
D→B B	3	C
D→C C	2	C
D→D D	0	D

ex: for the graph:



; find 'C' vectors if delays from C to B = 6
C to D = 3
C to E = 5.

⇒

dest router	dist	next router
A	11	B
B	6	B
C	0	C
D	3	D
E	5	E
F	8	B

⇒ trick: * fill x → x as '0' and 'x'.
* fill neighbours y as 'dist' and 'y'.
* cal for remaining routers using above method.

neighbors are D, B, E

$C \to B + B \to A = 11$
$C \to D + D \to F + F \to E + E \to A = 24$
$C \to E + E \to A = 12$

smallest is 11

$C \to D + D \to F = 13$
$C \to E + E \to F = 9$
$C \to B + B \to F = 8$
$C \to B + B \to A + A \to E + E \to F = 22$

smallest is 8

Use - Cases:

① LANs ② home networks ③ ISPs ④ Intranets ⑤ PPP networks.

Advantages: ① Simplicity ⇒ straightforward to implement & understand.
② low resource req. ⇒ min processing power & memory.
③ easy setup ⇒ min network knowledge.
④ automatic updates ⇒ share routing tables with neighbours at regular intervals.

Disad: ① Routing loops ⇒ circulation of packets indefinitely within the network due to wrong table entries.
② Slow Convergence ⇒ count-to-∞ prob can cause delays in route updates leading to routing loops.
③ limited scalibility ⇒ good for small / medium scale networks not for larger networks.

Count - to - infinity Problem: (fundamental limitation of D.V.R)

⇒ drawback of dist vector routing.
⇒ routers can continuously update their routing tables with wrong information leading to routing loops.
⇒ It occurs when there is a network topology change but the routing info does not propagate quickly / efficiently through the network and the routers keep circulating wrong distance vectors) values.
⇒ network instability.
⇒ increased network traffic.
⇒ delayed / slow convergence.

Convergence:

⇒ process of all routers reaching a consistent state where their routing tables reflect the current network topology accurately ⇒ Convergence.

⇒ settling of routes to best paths across the network ⇒ convergence

⇒ a set of routers in a network share the same topological information.

⇒ fast conv: D.V.R reacts quickly to +ve changes (ex: a new shorter path)

⇒ slow conv: D.V.R reacts slowly to −ve changes (ex: link failure) ⇒ leads to Count-to-∞ problem

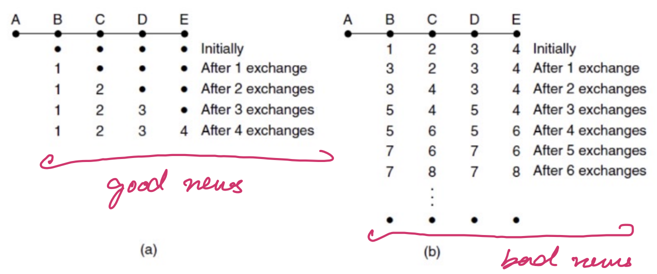


Figure 5-10. The count-to-infinity problem.

⇒ there are 2 cases if table is updated: (i) good news propagation (router comes back online)
(ii) Bad news propagation (router fails / link fails)

(i) * initially say router A is down and BCDE have recorded their delays to A as ∞.

* when A comes back online, router B is first to learn about it since it is directly connected to A so B updates its routing table to reflect that A is now 1 hop away. Other routers still believe A is down

* in next exchange, router C learns from B that A is reachable via B and a delay of 2 hops is updated by C.

* Same is done by D. This shows that good news spreads one hop per exchange.

* all routers would have updated their routing tables to reflect the new, shorter path to A

(ii) * say all routers BCDE have dists to A of 1, 2, 3 & 4 hops respectively.

B: 1 hop to A
C: 2 hops to A (via B)
D: 3 hops to A (via C)
E: 4 hops to A (via D)

assume.

* If A goes down or link b/w A to B is cut then B will no longer hear from A (ideally, B shd set its dist to A as ∞ in table).

* But C still believes there is a path to A through B with 2 hops.

* B, seeing that C claims to have a path to A with 2 hop incorrectly assumes that C has a valid, alternate route to A. So B updates its route as 3-hops to A via C.

* In next exchange, C sees that B has 3 hops route to A via B and updates route as 4 hops thinking it has a path to A via B.

* This continues with D & E also updating incorrect info on their tables

* The no. of hops keeps increasing even though A is un-reachable ⇒ Count-to-∞ prob.

Link- State Routing:

⇒ ARPANE: first wide area packet switch network, initially used D·V·R.
⇒ primary prob with D·V·R was slow convergence.
⇒ To address the probs of D·VR, ARPANET adopted link state routing in 1979.
⇒ It became widely used : * faster convergence.
* more reliable.

Ex:

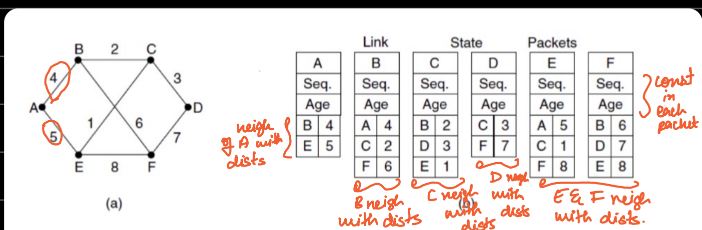


Figure 5-12. (a) A network. (b) The link state packets for this network.

step①:
⇒ a link state packet consists seq no., age, neighbours
⇒ Basically by flooding, info of each router is broadcasted in the network. Each router has global knowledge of other
step②:
⇒ with help of dijk alg we need to find the shortest path:
* smallest is 4, so repeat it.

	B	C	D	E	F	
A	4	∞	∞	5	∞	
AB	4	6	∞ (no direct edge)	5	10	(5 is smallest)
ABE	4	6	∞ (no direct edge)	5	10	(6 is smallest)
ABEC	4	6	9	5	10	(9 is smallest)
ABECD	4	6	9	5	10	(10 is smallest)

∴ shortest path is B C D E F: 4 6 9 5 10 //

⇒ How? 5 steps:

① discover its neighbours & their addresses: By sending HELLO packets to each of its neigh routers so that each router has info of the other initially itself.

② set link costs to each of its neighbours: * each link shd have a dist / cost metric for finding shortest paths
* the cost can be set automatically / configured by the network operator. The cost shd be inversely proportional to the bandwidth of the link.
* an ECHO packet is sent and is to be sent back immediately, this is to determine the delay by measuring the round trip of ECHO packet & div it by 2. This way the sending router can get an estimate of the delay.

③ Build link state packets: * each router shd build a packet containing all the data.
* each link state packet consists of i) identity of sender iii) age
ii) sequence no. iv) neighbours with cost to each neighbour.

④ distribute link state packets: * all routers must get all of the link state packets quickly & reliably.
* fundamental idea is to use flooding as diff topologies might be present & topology can be changing constantly.
* each packet has a seq no. to keep the flood in check.

* seq no. is incremented for each new packet sent. Whenever a new packet comes in, it is checked against the list of packets already seen. If it is new, it is forwarded on all the lines except the one it arrived on. If it is duplicate, it is discarded.
* if a packet with seq no lower than the highest one seen so far arrives, it is rejected.
* the age field is also decremented by each router along with incrementation of the seq no. to make sure no. packet lives for ∞ period of time. ex: a packet whose age is 0 is discarded.
* In short, whenever a link state packet comes in to a router for flooding, it is not queued for transmission immediately, it is put in a holding area to wait in case more links are coming up/going down. If another link state packet from the same source comes in b4 the 1st packet is transmitted, their seq nos are compared & if they are equal, duplicate is discarded. If they are diff, older one is thrown out. All link state packets are acknowledged to avoid errors.

⑤ Compute routes/shortest path to every router: * Shortest path is computed by dijk alg.

Use cases:

⇒ IS-IS link state protocol: Intermidiate System - Intermidiate System protocol.
⇒ used by many ISPs. ⇒ can carry info abt multiple network layer protocols at the same time.

⇒ OSPF (open shortest path first) link state protocol ⇒ OSPF cannot carry info about multiple network layer protocols at the same time.

⇒ large enterprise networks, ISP networks, cloud networks, MANs, military networks

Advantages:

1. fast convergence (use OSPF and IS-IS)
2. scalability (large)
3. efficient use of bandwidth.
4. adaptation to topology changes. (detect topology changes)
5. loop-free routing (due to dijk alg)

Disadvantages:

1. Complex Configuration
2. high resource req.
3. flooding overhead
4. high maintenance for large networks//
5. Synchronization delays//

Hierarchical Routing:

⇒ Size of routing tables can become a serious problem. As networks grow, it becomes necessary to adopt to hierarchial routing to ensure eff performance & scalibility.

⇒ as network becomes larger; mem consumption increases & CPU processing time increases with bandwidth.

⇒ In H.R, routers are divided into <u>regions</u> (areas). Each router knows all info within its own region but only limited info about other regions. This reduces the size of the routing tables.

⇒ for huge networks, two level hierarchy may be insufficient; it may be necessary to group regions into clusters; clusters into zones; zones into groups & so on....

Types:

① Intra-region routing:

within a region, routers know every info / routes etc. Routing is done with full knowledge of the network topology within that specific area.

② Inter-region routing:

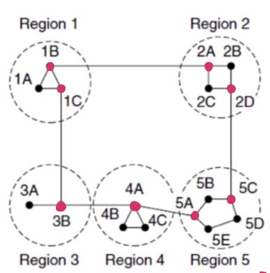
when sending packets to another region, router only needs to know the gateway router for the region, not the full path within the region. This reduces complexity of routing tables.

③ Two-level Hierarchy:

in many cases, H.R uses two levels. Routers are grouped into two regions & each region has routing info for other regions

④ Multi-level Hierarchy:

in extremely large networks, a two level hie might not be sufficient. The network may need to be further subdivided to cluster, zones & groups to reduce complex routing tables.



routers are div into regions : 1, 2, 3, 4, 5.

(a)

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

(b)

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

(c)

Figure 5-14. Hierarchical routing.

⇒ Basically 1A has info about 1B and 1C. since they belong to region 1. ||ly, 1C & 1B has info about all 3 routers in the region.

⇒ Remember that 1A does not have info about other regions. Hence, each router has info only about routers of the same region, not other regions.

⇒ To overcome this prob, we are using the <u>gateway routers</u>. A gateway router has info about other region's router too. For ex; 1B is a gateway router as it has info about 2A, 1C is a gateway router as it has info about 3B.

⇒ in the above ex: 1B, 1C, 2A, 2D, 3B, 4A, 5A, 5C are gateway routers / link b/w one region to another region in a network.

⇒ In the table for 1A: fig(b):

* hops ⇒ routers passed by to reach dest.
⇒ do not consider the dest router as a hop.

Full table for 1A

	Dest.	Line	Hops
1	1A	–	–
	1B	1B	1
	1C	1C	1
2	2A	1B	2
	2B	1B	3
	2C	1B	3
	2D	1B	4
3	3A	1C	3
	3B	1C	2
4	4A	1C	3
	4B	1C	4
	4C	1C	4
5	5A	1C	4
	5B	1C	5
	5C	1B	5
	5D	1C	6
	5E	1C	5

⇒ we have 2 columns: line & the hops.

⇒ dest has names of all routers region wise.

⇒ line represents the link / edge / via which router the packet is being sent.

⇒ wrt 1A is the source router. Say 1B is the dest router, we will be transmitting the packet via 1B only with only 1 hop. IIIy, 1A → 1C we would be passing by 1C only with 1 hop.

⇒ Say dest is 2A; from 1A→2A, 1A will send packet to 1B i.e 1 hop; 1B will send packet to 2A i.e 1 hop. So a tot of 2 hops are req from 1A→2A and the line will be 1B.

⇒ for 1A→2B we can go through 1B or 1C but will choose the one that req less hops so we will choose 1B route with 3 hops.

Hierarchical table for 1A

	Dest.	Line	Hops
1	1A	–	–
	1B	1B	1
	1C	1C	1
others	2	1B	2
	3	1C	2
	4	1C	3
	5	1C	4

always check for the nearest router of the region ex: 3B //

⇒ This is the hierarchial table for 1A. In this we will be representing regions. Source router is 1A.

⇒ we note down the hops for routers of the region of 1A.

⇒ Then we note down the hops for the 2, 3, 4, 5 regions.

Advantages:

① Scalability
② reduced routing table size.
③ less complex tables.
④ efficient use of bandwidth.
⑤ improved CPU performance
⑥ eff mem usage.

Disadvantages:

① Loss of optimal routes.
② Complex Configuration.
③ additional overhead for large hierarchies.

Broadcast Routing: (watch "Sudhakar Atchala" yt video)

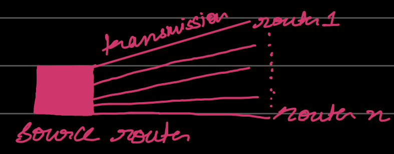
⇒ It is the process of sending a packet from 1 source to all other hosts in a network. It is commonly used for services that need to deliver data to many machines in the network. ex: Stock updates.
weather reports.
Streaming programs.

⇒ Sending a packet to all destinations is called broadcasting.
⇒ methods:

① Simple Broadcast (point-to-point trans): The source sends a seperate copy of packet to each destination.

ad * easy to implement, straightforward.
* no need for special routing algs.

disad * insuff bandwidth due to high redundancy.
* req the source to have a complete list of all dests.
* slow due to individual packet trans to all dests.



② Multi-Destination Routing: single packet is sent containing list of dests. Router checks which output lines are needed & generates copies of the packet for each req link.

ad * eff bandwidth as packets are shared amg dests that share the same route.
* reduces redundancy by partitioning the dest set among output lines.

disad * source still needs to know all the dest.
* router has to perform extra work ⇒ split the dest set & generate copies of packets.

③ Reverse path forwarding (RPF): when a broadcast packet arrives at a router, it checks to see if packet has arrived via the optimal path (i.e the path used to send packets to source). If yes, the router forwards the packet to all other links. If not, packet is discarded as a duplicate.

ad: * simple & elegant solⁿ.
* avoids sending unnecessary duplicates by discarding non-optimal packets
* sends each packet only once per dirᶜ (reducing redundant transmission).

disad: * may still generate some duplicates if paths deviate from optimal route.

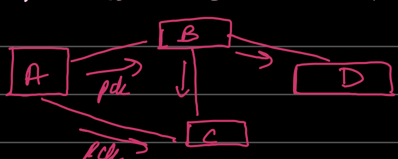
④ Spanning Tree Broadcast: a spanning tree, which is a subset of the network containing all routers but no loops is used. Broadcast packets are forwarded only along the edges of the Spanning Tree. aka sink tree.

ad: * min no. of packets req for broadcast.
* eliminates unnecessary re-transmissions by using a loop-free structure.

disad: * req routers to know the spanning tree strucᶜ.
* might not always provide most optimal route for every dest.

⇒ for large networks RPF and S.Ts Broadcasting are preferred.

⑤ Flooding: flooding is also a type of broadcasting method. We can use this method to overcome the prob of point-to-point redundancy (Simple Broadcast)



A will send packets to B & C and B will send it to C & D. In C we got a dup packet as it ald has a packet from A. (disadvantage of flooding: many dup packets are generated)

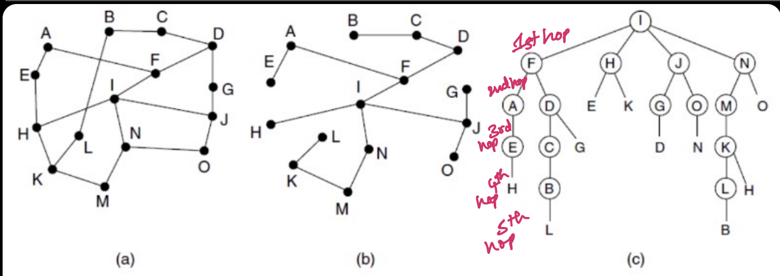
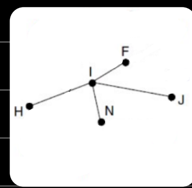


Figure 5-15. Reverse path forwarding. (a) A network. (b) A sink tree. (c) The tree built by reverse path forwarding.

⇒ assume we are starting at 'I'. From 'I' we can travel to F, N, H, J.

⇒ draw the links to the neighbours 1st like this: 

for (a).

⇒ now draw the links from each neighbour by avoiding the loops. as shown in (b).

⇒ from F we can join D, C, B and A and E, IIIly for others.

⇒ we can draw many forms of spanning trees but we must visit all the routers.

⇒ The tree can also be called "sink tree".

⇒ To build a RPF tree as shown in (c) by the diag (a):

* On 1st hop, I sends packets to F, H, J, N Each of these arrive on preferred path to I (sink tree), hence indicated by a circle.
* On 2nd hop, 8 packets are generated (two by each of the routers of 1st hop). wherein five are on preferred link / sink tree so we circle them.
* from F we can go to A or D. IIIly for N, J, H.
* On 3rd hop, only 3 are on preferred line & others are duplicates.
* after 5 hops and 24 packets, broadcasting terminates compared with 4 hops and 14 packets on preferred path / sink tree.
* advantage is the router shd know only how to reach all dest not the seq no. or without any knowledge / info / list on the route.

Use Cases:

1. Software update announcements.
2. Service announcements.
3. routing updates //

Multicast – Routing:

⇒ sending message from 1 source to multiple receivers in a network.

⇒ unlike broadcasting where packets are sent to all the nodes or unicasting where packets are only sent to one node.

⇒ M.C is used when a group of nodes needs the message but is relatively small compared to the entire network

⇒ A multicast group consists of the set of receivers interested in receiving specific packets. These grps are identified by multicast address.
⇒ multicast routing is used to route messages to multiple dests (members of a multicast group) in the network.
⇒ Techniques:

① Multicast Spanning Tree: builds a tree from source to all group members. The broadcast S·T is pruned to exclude links that do not lead to group members. Pruning the S·T reduces unnecessary links & results in a more efficient tree for that group.
ad: * efficient as it eliminates packet trans to un-interested routers.
* reduces bandwidth usage.
disad: * Tree must be recalculated if group membership changes.
* high maintenance / complex main. of trees for diff groups

② Dist Vector Multicast routing: DVMRP uses reverse path forwarding. It is an example of multicast routing protocol. When a router that does not serve any group members receives a multicast packet, it sends a PRUNE message to upstream routers to stop sending multicast messages for that group. This process prunes the multicast tree over time / reduced tree form.
ad: * effi spanning trees that only use links to reach grp members.
* adapts dynamically by pruning unnecessary links.
disad: * reqs significant processing, especially in large networks.
* may generate many PRUNE messages in large / dense networks //

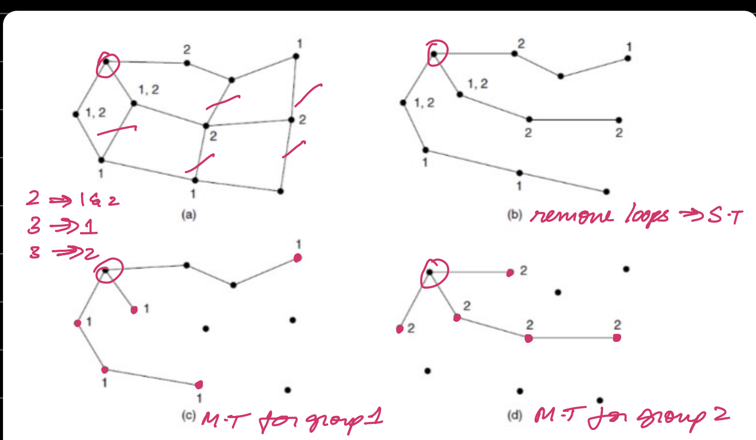


Figure 5-16. (a) A network. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1. (d) A multicast tree for group 2.

⇒ 1 and 2 are two groups in network ⓐ.
⇒ some routers are attached to hosts that belong to one or both of these groups.
⇒ ⓑ: is the S·T for the leftmost router that can be used for broadcast but not multicast
⇒ ⓒ: all links that do not lead to hosts that are members of group 1 have been removed. This is a M·C·S·T for the leftmost router to send to group 1. This is more eff than the broadcast tree as there are 7 links and not 10. ⓓ: shows tree after pruning for group 2. This is also efficient with only 5 links. Therefore, diff M·C groups have diff S·Ts.

Advantages: ① Scalability ⇒ many receivers comm without overhead.
② eff bandwidth ⇒ reduces traffic.
③ flexibility ⇒ supports dynamic grp management allowing hosts to join or leave grps as needed.

Disad:

1. Complex grp management: which router belongs to which group.
2. more processing time: group memberships shd be maintained, increasing processing requirements.
3. resourse-intensive: managing PRUNE messages & M.C tables updating etc..

Uses:

1. Gaming (multiplayer online games).
2. Live Video Streaming (news, sports etc)
3. IPTV (int protocol television).
4. video conferencing.

Broadcast Routing	Multicast routing
⇒ sends data to all nodes in the network	⇒ sends data to specific group of receivers.
⇒ data is sent to un-intereed nodes also, less efficient.	⇒ data is sent to only the interested nodes, more efficient.
⇒ higher overhead due to unnecessary packet distribution.	⇒ less overhead as packets are sent only to group members.
⇒ Scales poorly as network size increases.	⇒ Scales better for large group members.
⇒ flooding, RPF, Simple, S.Ts.	⇒ DVMRP, PIM (protocol independent multicast)
⇒ no control of traffic: sent to all nodes	⇒ more controlled traffic: only sent to defined members of the group.
⇒ less security	⇒ more security.
⇒ Bandwidth is wasted.	⇒ Bandwidth is used effectively.
⇒ (use cases diff)	