UNIT 2 PART 1

Network Layer Design Issues: Store-and-Forward Packet Switching, Services Provided to the Transport Layer, Implementation of Connectionless Service, Implementation of Connection-Oriented Service, Comparison of Virtual-Circuit & Datagram Subnets.

Network Layer

- The network layer is mainly responsible for transmitting data packets from the source and delivers them to the destination.
- This layer decides the route for packets by using **Routing Algorithms**.

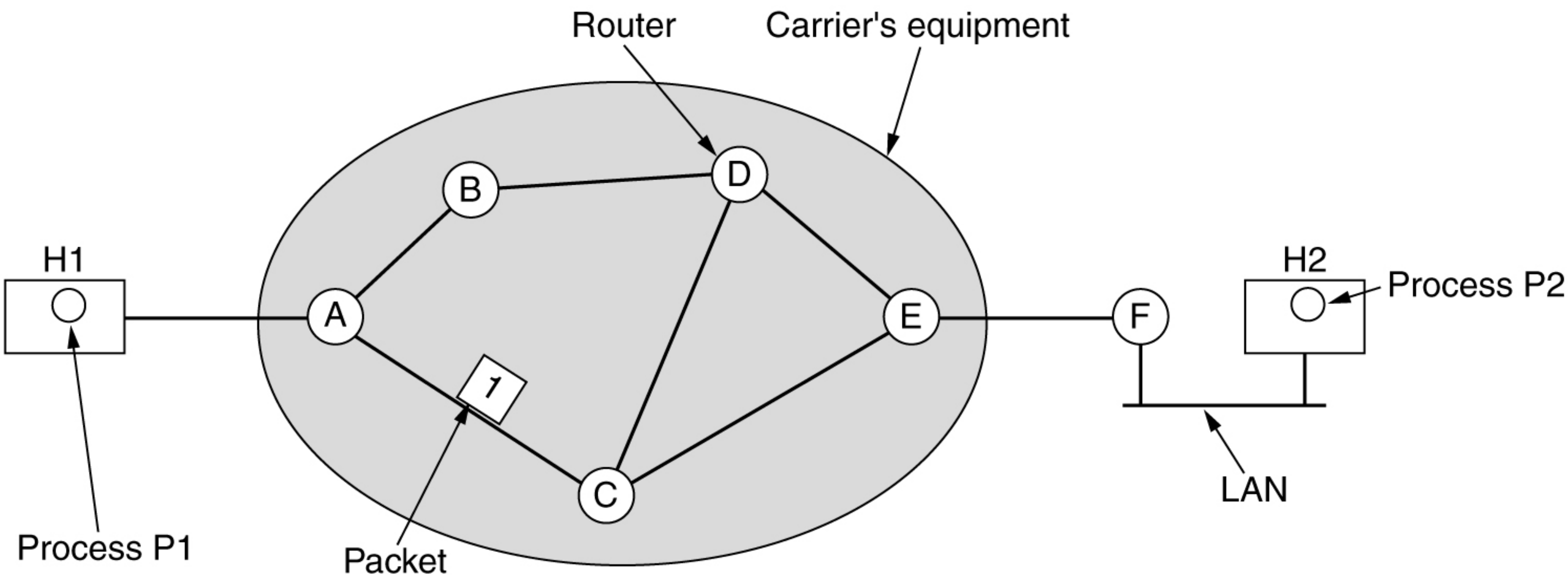
Network Layer Design Issues

- The network layer design issues include the service provided to the transport layer and the internal design of the network.

Store-and-Forward Packet Switching

- The major components of the network are the ISP's equipment (routers connected by transmission lines), shown inside the shaded oval, and the customers' equipment, shown outside the oval.
- Host H1 is directly connected to one of the ISP's routers A, as shown in below figure.

Store-and-Forward Packet Switching



The environment of the network layer protocols

Store-and-Forward Packet Switching

- In above figure, A host H1 send a packet to the nearest router, either on its own LAN or over a point-to-point link to the ISP.
- The packet is stored there until it has fully arrived and the link has finished its processing by verifying the checksum.
- Then it is forwarded 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.**

Services Provided to the Transport Layer

- The network layer provides services to the transport layer at the network layer/transport layer interface.
- An important question is precisely what kind of services the network layer provides to the transport layer.
- The services need to be carefully designed with the following goals in mind:

1. The services should be independent of the router technology.
2. The 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 numbering plan, even across LANs and WANs.

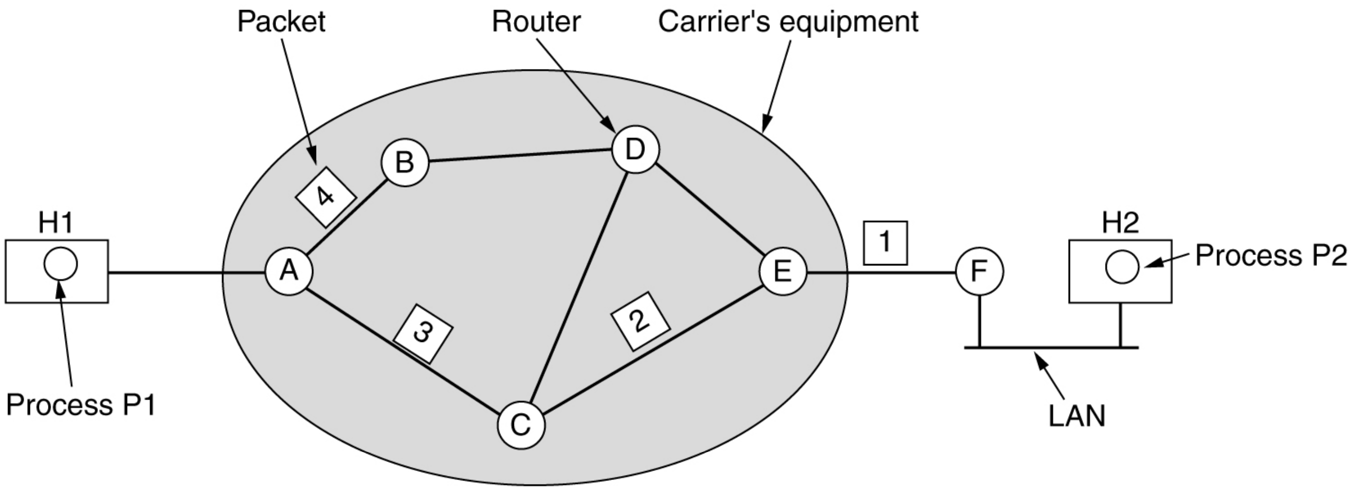
Services Provided to the Transport Layer

- Based on the connections there are two types of services provided:
- **Connectionless** – The routing and insertion of packets into subnet is done individually. No added setup is required.
- **Connection-Oriented** – Subnet must offer reliable service and all the packets must be transmitted over a single route.

Implementation of Connectionless Service

- If connectionless service is offered, packets are injected into the network individually and routed independently of each other.
- No advance setup is needed.
- In this context, the packets are frequently called **datagrams** (in analogy with telegrams) and the network is called a **datagram network**.

Implementation of Connectionless Service



A's table

initially

Dest.	Line
A	–
B	B
C	C
D	B
E	C
F	C

later

Dest.	Line
A	–
B	B
C	C
D	B
E	B
F	B

C's table

Dest.	Line
A	A
B	A
C	–
D	D
E	E
F	E

E's table

Dest.	Line
A	C
B	D
C	C
D	D
E	–
F	F

Routing within a diagram subnet

Implementation of Connectionless Service

- Let us now see how a datagram network works.
- Suppose that the process P1 in above figure has a long message for P2.
- It hands the message to the transport layer, with instructions to deliver it to process P2 on host H2.
- The transport layer code runs on H1 prepends a transport header to the front of the message and hands the result to the network layer.
- Let us assume for this example that the message is four times longer than the maximum packet size, so the network layer has to break it into four packets, 1, 2, 3, and 4, and send each of them in turn to router *A.*
- At this point the ISP takes over.

Implementation of Connectionless Service

- **Every router has an internal table** telling it where to send packets for each of the possible destinations.
- Each table entry is a pair consisting of a destination and the outgoing line to use for that destination. Only directly connected lines can be used.
- For example, in above figure, A has only two outgoing lines—to B and to C—so every incoming packet must be sent to one of these routers, even if the ultimate destination is to some other router.
- A's initial routing table is shown in the figure under the label ''initially.''

Implementation of Connectionless Service

- At A, packets 1, 2, and 3 are stored briefly, having arrived on the incoming link and had their checksums verified.
- Then each packet is forwarded according to A’s table, onto the outgoing link to C within a new frame.
- Packet 1 is then forwarded to E and then to F.
- When it gets to F, it is sent within a frame over the LAN to H2.
- Packets 2 and 3 follow the same route.

Implementation of Connectionless Service

- However, something different happens to packet 4.
- When it gets to A it is sent to router B, even though it is also destined for F.
- For some reason, A decided to send packet 4 via a different route than that of the first three packets.
- Perhaps it has learned of a traffic jam somewhere along the ACE path and updated its routing table, as shown under the label ‘‘later.’’
- The algorithm that manages the tables and makes the routing decisions is called the **routing algorithm.**

Implementation of Connection-Oriented Service

- If connection-oriented service is used, a path from the source router all the way to the destination router must be established before any data packets can be sent.
- This connection is called a **VC (virtual circuit),** in analogy with the physical circuits set up by the telephone system, and the network is called a **virtual-circuit network.**

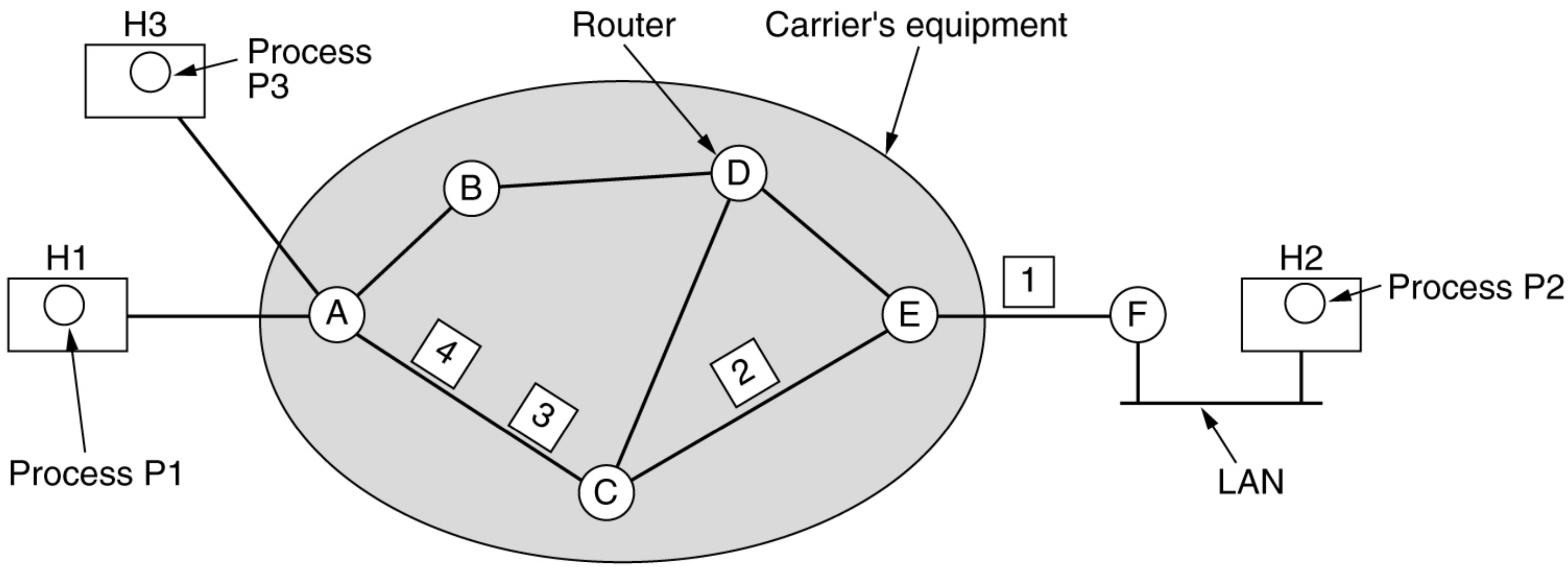
Implementation of Connection-Oriented Service

- In connection oriented service, when a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers.
- That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works.
- When the connection is released, the virtual circuit is also terminated.

Implementation of Connection-Oriented Service

- With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.
- In virtual Circuit subnets, each VC is numbered and every router maintains a table.
- Since all the packets of the given virtual circuit follow the same route through the subnet.
- Each packet’s header contains virtual circuit number.
- The router uses this number to forward the packets on correct output line.

Implementation of Connection-Oriented Service



A's table

In		Out	
H1	1	C	1
H3	1	C	2

C's table

A	1	E	1
A	2	E	2

E's table

C	1	F	1
C	2	F	2

Routing within a virtual-circuit subnet

Implementation of Connection-Oriented Service

- As an example, consider the situation shown in above figure.
- Here, host H1 has established connection 1 with host H2.
- This connection is remembered as the first entry in each of the routing tables. The first line of A’s table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to router C and given connection identifier 1.
- Similarly, the first entry at C routes the packet to E, also with connection identifier 1.

Implementation of Connection-Oriented Service

- **Now let us consider what happens if H3 also wants to establish a connection to H2.**
- It chooses connection identifier 1 (because it is initiating the connection and this is its only connection) and tells the network to establish the virtual circuit.
- This leads to the second row in the tables.
- Note that we have a conflict here because although A can easily distinguish connection 1 packets from H1 from connection 1 packets from H3, C cannot do this.
- For this reason, A assigns a different connection identifier to the outgoing traffic for the second connection.
- Avoiding conflicts of this kind is why routers need the ability to replace connection identifiers in outgoing packets.
- In some contexts, this process is called **label switching.**

Comparison of Virtual-Circuit and Datagram Subnets

Issue	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
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

UNIT 2 PART 2

Routing Algorithms: The Optimality Principle, Shortest Path Routing, Flooding, Distance Vector Routing, Link State Routing, Hierarchical Routing

Routing Algorithms

- The **routing algorithm** is that part of the network layer software called **routing protocol** that decides the path to be used for routing a packet from source router to destination router.
- Given a set of routers connected with links, a **routing algorithm** determines best path for a packet from source router to destination router.
- A path with least cost is referred to as the **best path**.
- Routing algorithms has certain properties that include: **correctness, simplicity, robustness, stability, fairness, optimal and efficiency**.

Classification

- Routing algorithms can be classified into two categories. They are:

1. Non-Adaptive Routing Algorithms
2. Adaptive Routing Algorithms

Non-Adaptive Routing Algorithms

- These algorithms are **static routing algorithms**.
- In these algorithms the route to be used for routing packets from source router to destination router is decided in **advance**, offline, and downloaded to the routers when the network is booted.
- Nonadaptive algorithms do **not** base their routing decisions on any measurements or estimates of the current topology and traffic.
- There are two types of static routing algorithm. They are

1. Shortest path Routing Algorithm
2. Flooding

Adaptive Routing Algorithms

- Adaptive routing algorithms are the dynamic algorithm, routers build their routing tables dynamically.
- Adaptive Routing algorithms change their routing decisions to reflect changes in the topology, and sometimes changes in the traffic as well.
- These dynamic routing algorithms differ in where **they get their information** (e.g., locally, from adjacent routers, or from all routers), when they **change the routes** (e.g., when the topology changes, or every ΔT seconds as the load changes), and **what metric is used for optimization** (e.g., distance, number of hops, or estimated transit time).
- There are two types of dynamic routing algorithm. They are

1. Distance vector routing algorithm
2. Link state routing algorithm

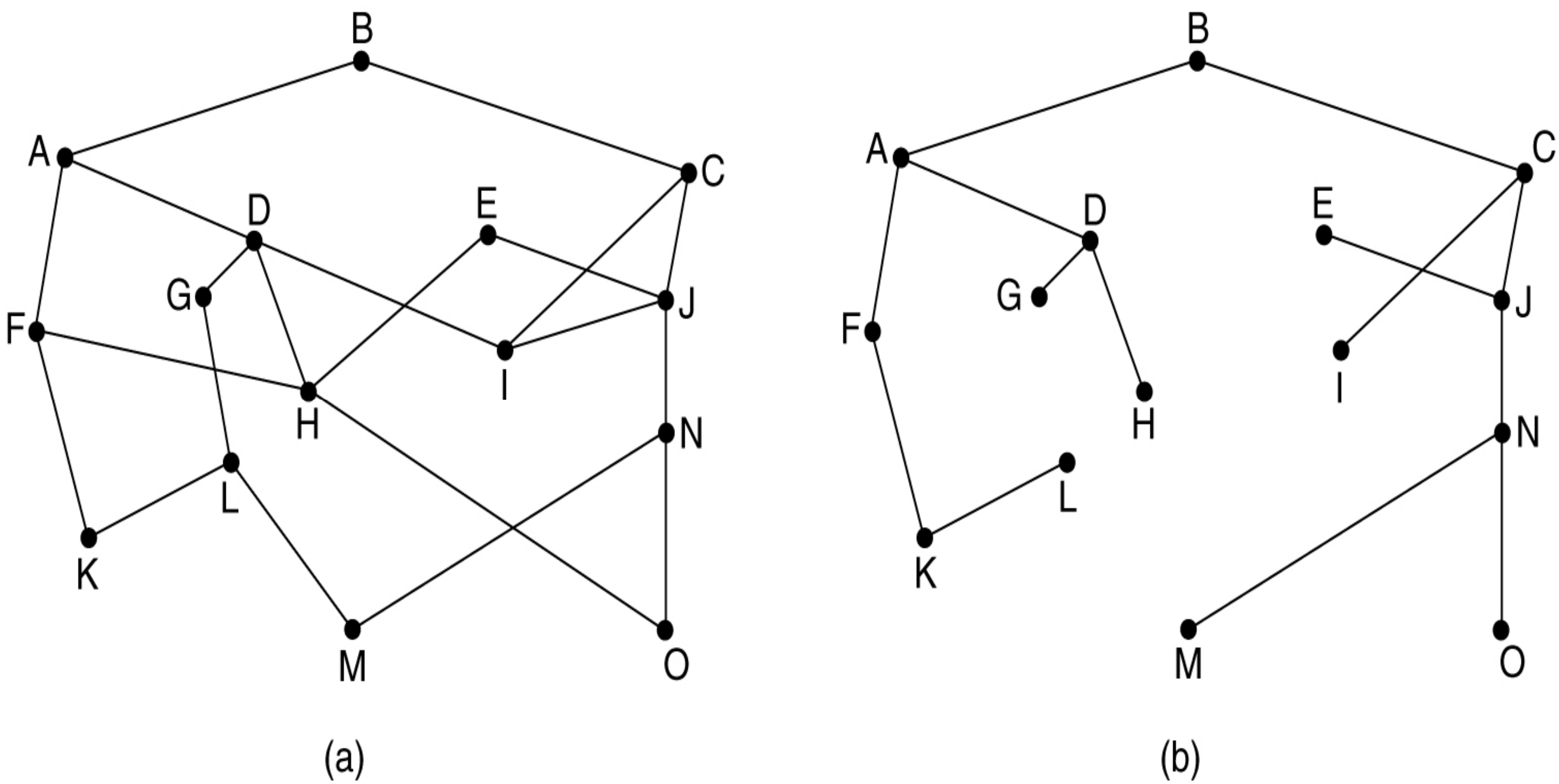
The Optimality Principle

- The optimality principle says that optimal routes can be made without the knowledge of the network topology or traffic load. It can be stated as follows.
- It states that if router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route. To see this, call the part of the route from I to J is r1 and the rest of the route is r2.
- If a route is better than r2 existed from J to K, it could be concatenated with r1 to improve the route from I to K, contradicting our statement that r1+r2 is optimal.

The Optimality Principle

- As a direct consequence of the optimality principle, we can see that the set of optimal routes from all sources to a given destination form a tree rooted at the destination.
- Such a tree is called a **sink tree** and is illustrated in figure(b), where the distance metric is the number of hops.
- Note that a sink tree is not necessarily unique; other trees with the same path lengths may exist.
- The goal of all routing algorithms is to discover and use the sink trees for all routers.

The Optimality Principle



(a) A subnet. (b) A sink tree for router B.

The Optimality Principle

- Since a sink tree is indeed a tree, it does not contain any loops, so each packet will be delivered within a finite and bounded number of hops.
- Links and routers can go down and come back up during operation, so different routers may have different ideas about the current topology.

Shortest Path Routing

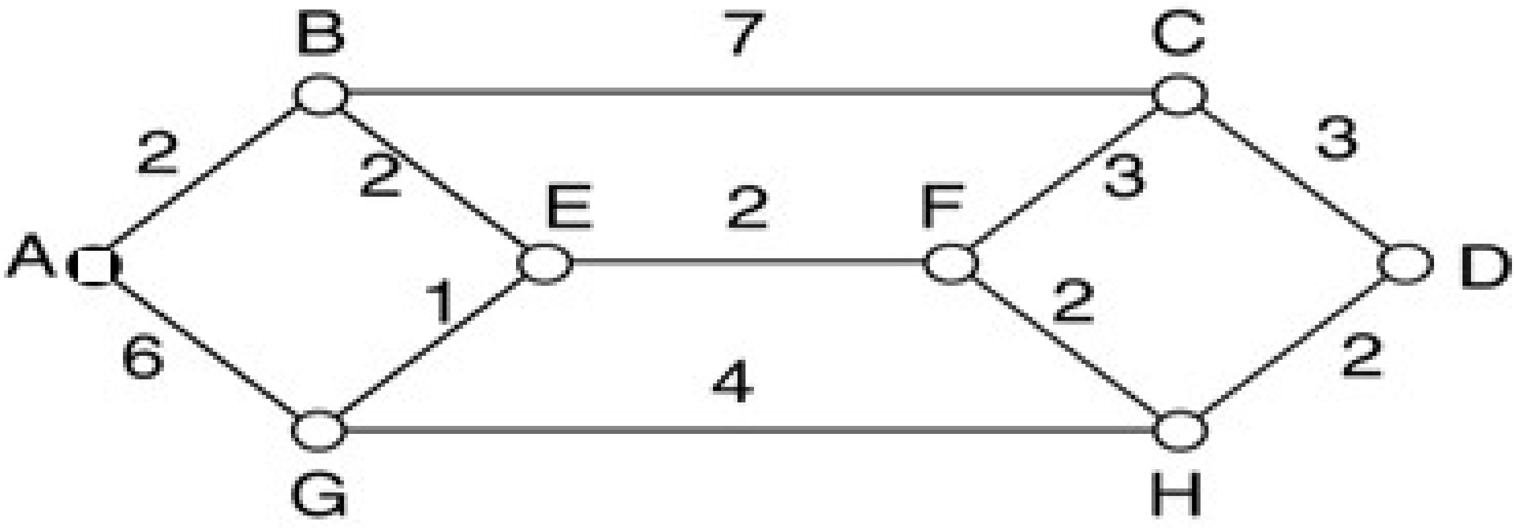
- The shortest path routing algorithm is a static routing algorithm.
- The algorithm builds a graph of the subnet where each node represents a router and each arc represents a link or communication line between two routers.
- The algorithm chooses a router between a pair of nodes by finding the shortest path between them.

Shortest Path Routing

- The metric used for measuring the length of a path may be either the number of hops (or) physical distance (or) cost.
- Several algorithms are used for computing the shortest path between any two nodes.
- The one is Dijkstras Algorithm for finding the shortest path.
  - The algorithm works by labeling each node with its distance from source node current node along the best known path.
  - Initially, all the nodes are labeled with infinity as paths are unknown. Initially, all the labels are tentative and made permanent on discovering that the label represents a shortest path from the source node to the current node.

Shortest Path Routing

- Consider the subnet as shown below figure:



- In this graph, each arc is labeled with distance. If we want to find the shortest path from router **A** to router **D** then,

Shortest Path Routing

1. Initially the algorithm marks the node **A** as permanent. Now **A** becomes the working node as shown in figure(a) indicated by filled circle.
2. Next the nodes adjacent to the working node i.e **A** are **B** and **G**.
3. The adjacent nodes are labeled with a pair of values. In given example **B** and **G** are relabeled as **(2, A)** and **(6, A)**. The nodes distance to **A** and the node from which this distance is possible to reconstruct the path later.

Shortest Path Routing

4. Finally the algorithm scans the graph for tentative nodes to make a node with smallest label as permanent. In above example **B** has the smallest path than **G** so it is made permanent.
5. The new permanent node becomes the new working node. This **B** becomes the new working node.

- The algorithm repeats above steps with the node **B**. the algorithm stops till reaches the desired destinations i.e **D** (router)
- The following steps are used in computing the shortest path from A to D as shown in below figure.

Shortest Path Routing

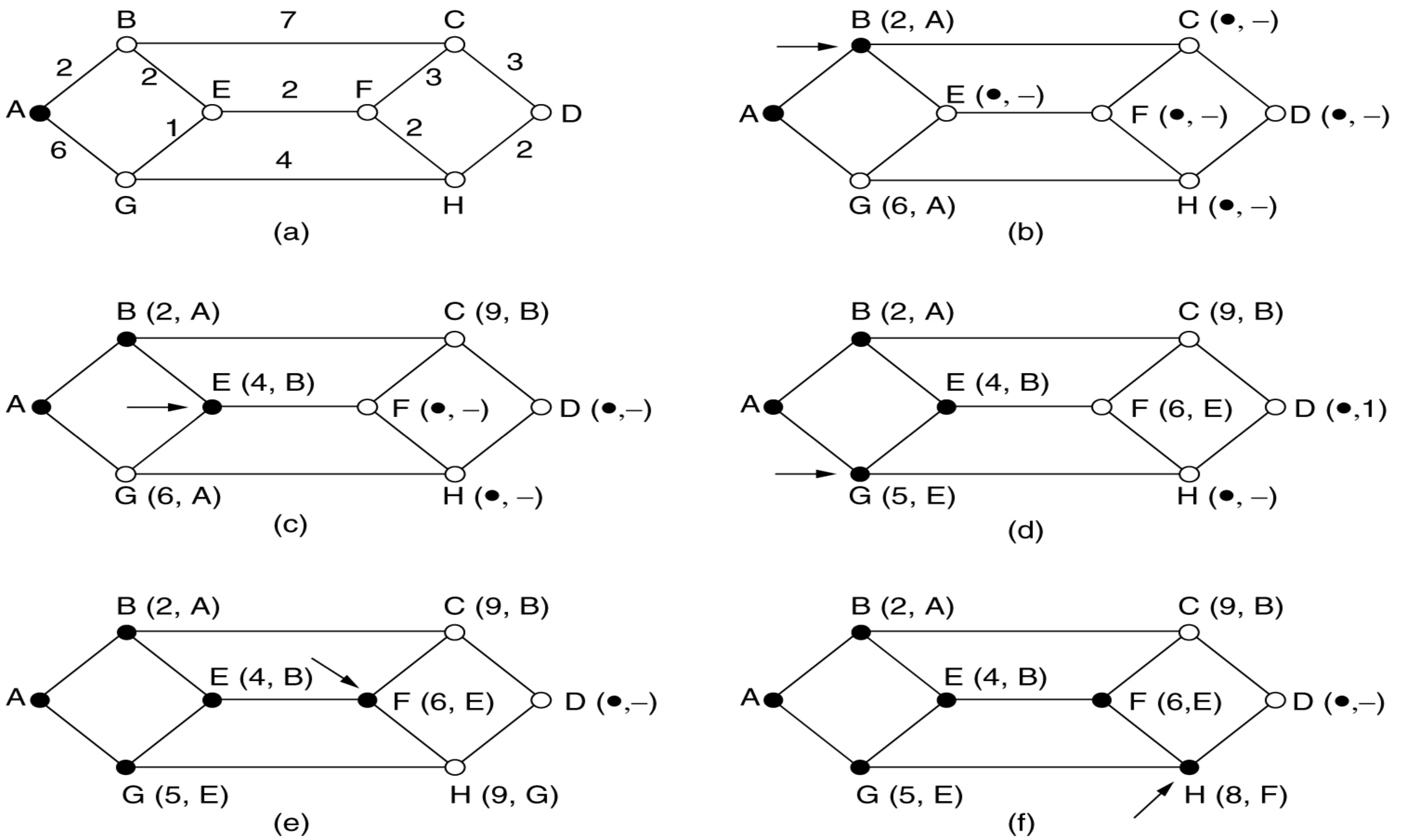
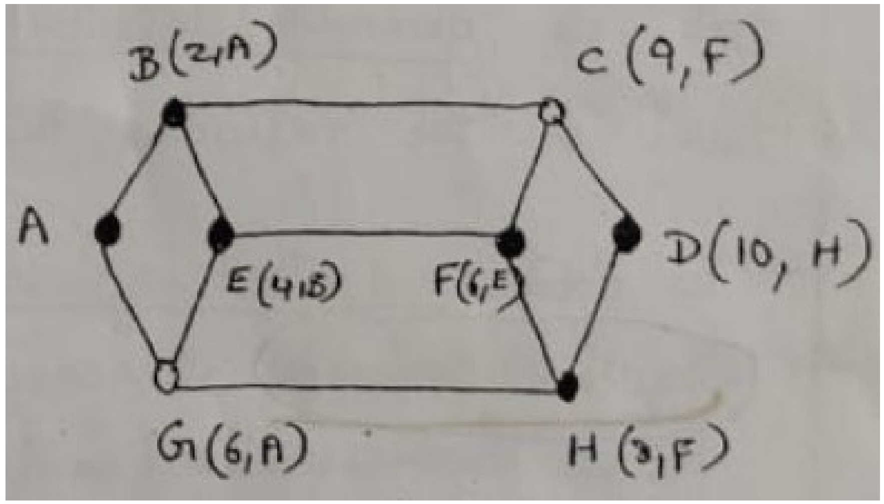
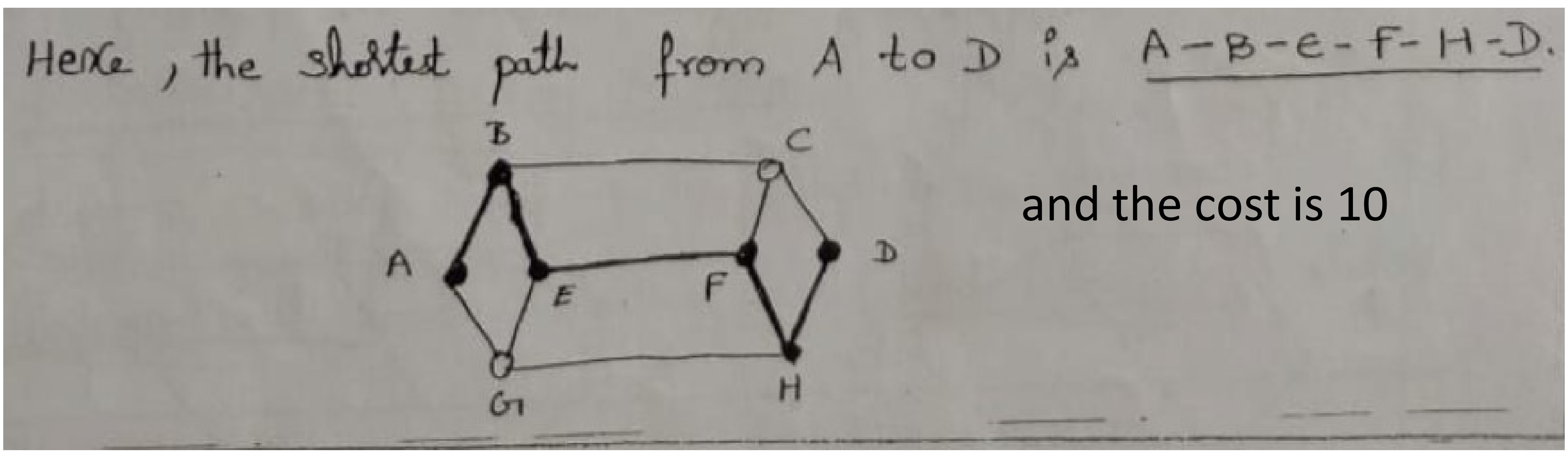


Figure: The first 5 steps used in computing the shortest path from A to D. The arrows indicate the working node.

Shortest Path Routing



(g)



and the cost is 10

Shortest Path Routing

- The **drawback** with this approach (shortest path routing algorithm) is that, it requires prior information about the entire network topology which becomes very difficult.

Flooding

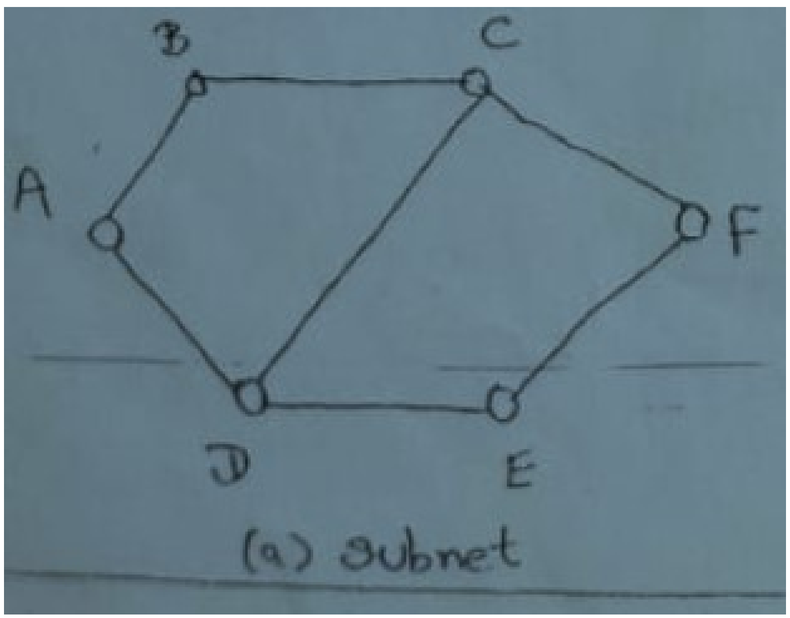
- Another static algorithm (non adaptive) is flooding, in which the routers sends an incoming packet to every outgoing line except the line on which it has arrived. This technique is called **flooding.**
- This approach does not require any prior information regarding network topology. The main problem with this approach is that it generates infinite number of duplicates packets travel through different paths.

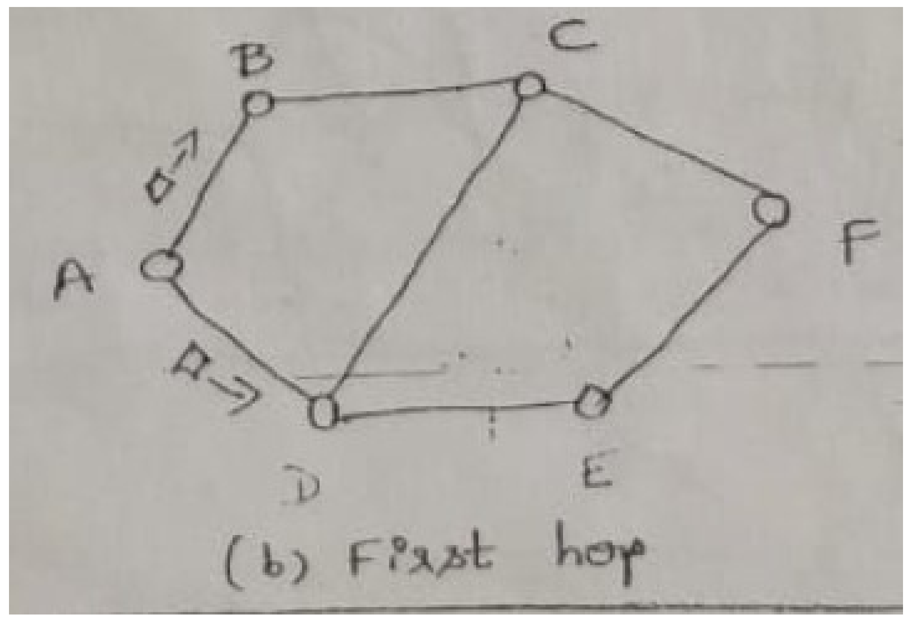
Flooding

- To prevent duplicate packets, the source router put a sequence number in each packet it receives from its hosts.
- Each router needs a list that source router telling which sequence numbers present at the source have already been seen.
- If an incoming packet is on the list, it is not flooded

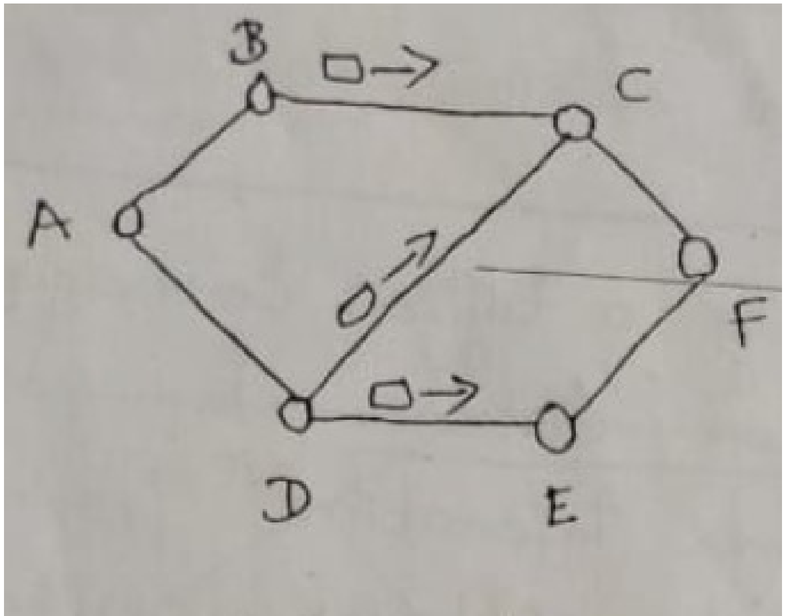
Flooding

- Example:- The following figure shows the flooding process for the subnet:

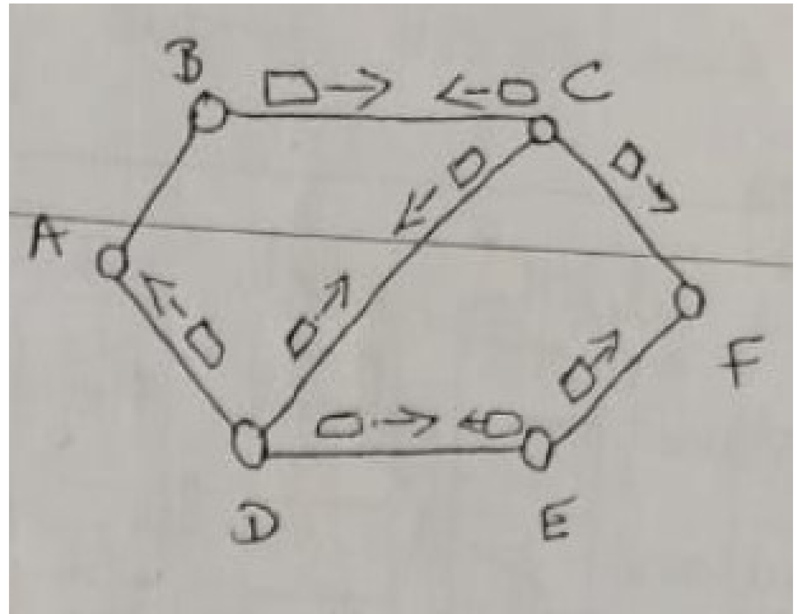




Flooding



(c) Second hop



(d) Third hop

Flooding

- In **selective flooding algorithm**, routers send every incoming packet only on those outgoing lines that lead to the right path rather than sending on every outgoing lines.

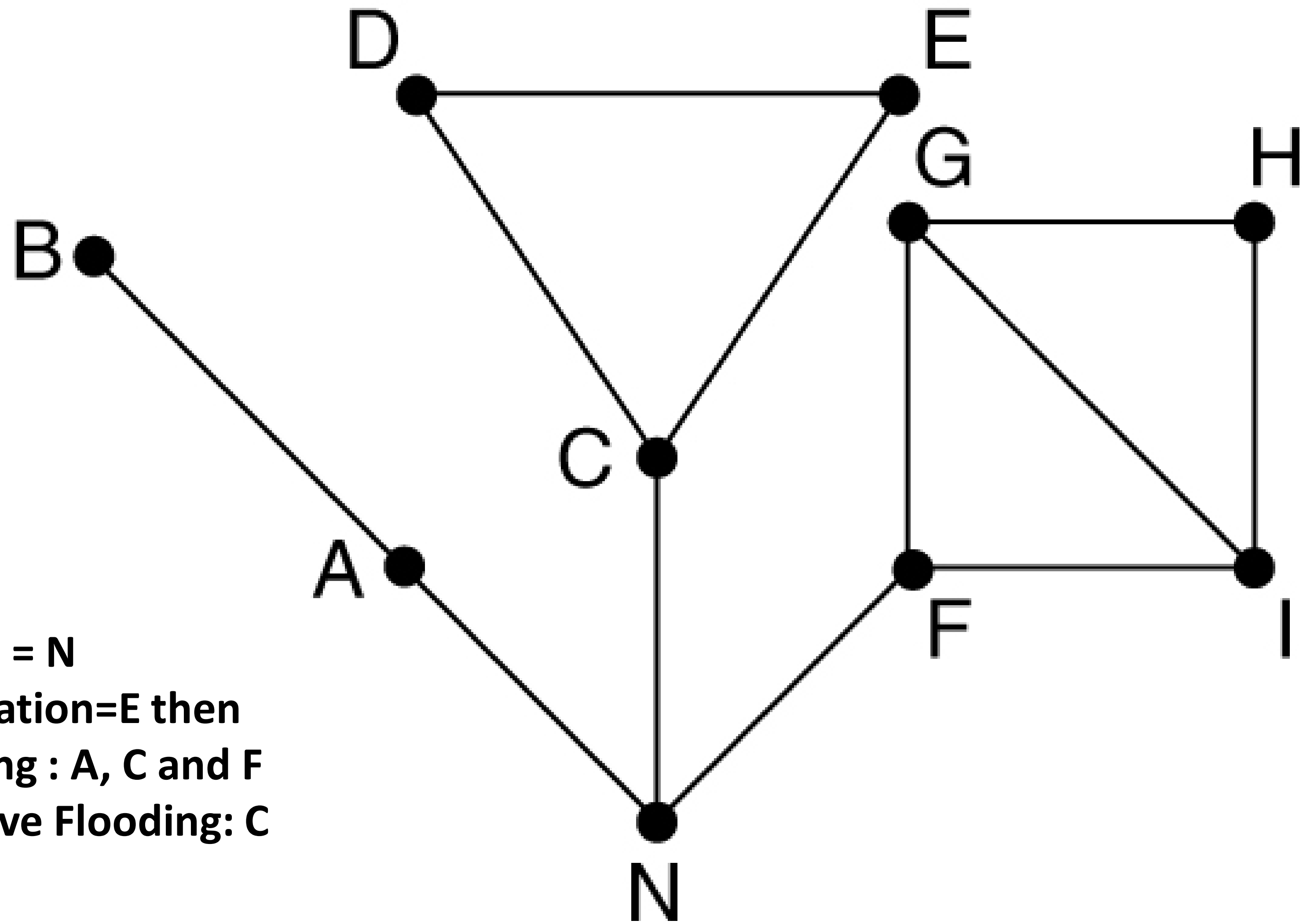
Flooding

- **Applications**
  - Flooding is used in military applications
  - It is used in distributed database applications
  - It is used in wireless networks

Flooding

- **Drawbacks of flooding**
  - It generates too many duplicate packets
  - To identify duplicate packets, each router need to store an identifier for every packet it has received
  - It consumes too much bandwidth
  - It increases the maximum traffic load placed on the network

Example



Source = N
Destination=E then
Flooding : A, C and F
Selective Flooding: C

Distance Vector Routing Algorithm

- This algorithm is a **dynamic routing algorithm**.
- This algorithm sometimes called as **Bellman-Ford routing algorithm** and **Ford-Fulkerson algorithm**.

Distance Vector Routing Algorithm

- In this algorithm, each router maintains a table containing the information about the best known distance to each destination.
- Each router receives some information from one or more of its directly attached neighbors, performs a calculation, and then distributes the results of its calculation back to its neighbors.

Distance Vector Routing Algorithm

- The routing table entry consists of two parts: **The preferred outgoing line to use for that destination** and **An estimate of the distance to that destination**.
- The metric used for routing could be the number of hops, the queue length (or) time delay in milliseconds.
- The tables are updated by exchanging information to directly connected neighbors.

Distance Vector Routing Algorithm

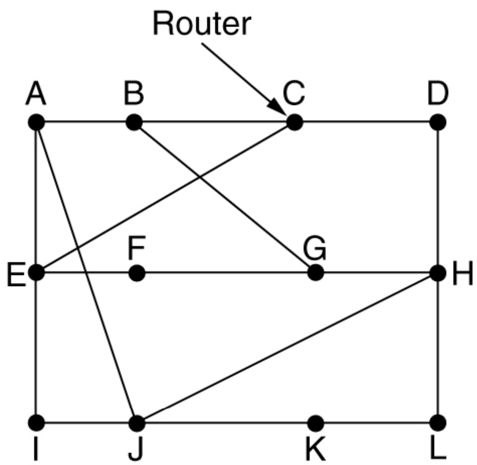
- If the delay metric is used then the router knows the time delay to reach each of its neighbors.
- After every T ms routers exchange their vectors update their tables with newly estimated delay to each destination.

Distance Vector Routing Algorithm

- Assume that the estimated delay of router **X** to reach the router **I** is xi. The router knows delay to reach the router **X** itself is ‘m’ ms, then the delay to reach to router **I** via **X** is (xi+m) ms.
- The router calculates the estimated delay for each neighbor and chooses the best estimated delay which is the lowest value. The router uses this delay and its corresponding line in its new routing table.
- Old routing table is not used in the calculation.

Distance Vector Routing Algorithm

For example, consider the following subnet as shown below:



(a)

To	A	I	H	K	New estimated delay from J	Line
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	New routing table for J	

Vectors received from J's four neighbors

(b)

(a) A subnet. (b) Input from A, I, H, K, and the new routing table for J.

Distance Vector Routing Algorithm

- In above figure, the fig (a) shows a network.
- The first four columns of fig (b) show the delay vectors received from the neighbors of router J.
- A’s delay vector shows that it can reach to router B in 12-msec, to router C in 25-msec, to router D in 9-msec.
- Suppose J’s estimated delay to its neighbors, A, I, H, and K, are 8, 10, 12, and 6 msec.

Distance Vector Routing Algorithm

- Now, consider how the router J uses the distance vector routing to compute its new router G.
- Router J knows its delay to its neighbor A is 8 ms and A can reach to G is 18 ms. The total delay is 8+18=26 ms
- Similarly, it computes the delay to G via I, H, K as 41(31+10), 8(6+12), 37(31+6) ms.

Distance Vector Routing Algorithm

- The best estimated delay is 18 ms which is possible if J forward packets via H. So, J makes an entry in its routing table the delay to G as 18 ms.
- Similarly the same calculation is used to calculate the delay to each destination using new routing table as shown above in last column.

Distance Vector Routing Algorithm

- In Brief

$J \xrightarrow{8} A \xrightarrow{18} G = 26$ ms

$J \xrightarrow{10} I \xrightarrow{31} G = 41$ ms

$J \xrightarrow{12} H \xrightarrow{6} G = 18$ ms (lowest value)

$J \xrightarrow{6} K \xrightarrow{31} G = 37$ ms

Distance Vector Routing Algorithm

- In Distance vector routing, routers identify the best path to reach the destination from each neighbor.
- It depend upon information from their adjacent neighbors to calculate and collect route information. They pass copies of routing table to their neighboring routers and get distance vectors. The updates on routing are performed step-by-step from router to router.

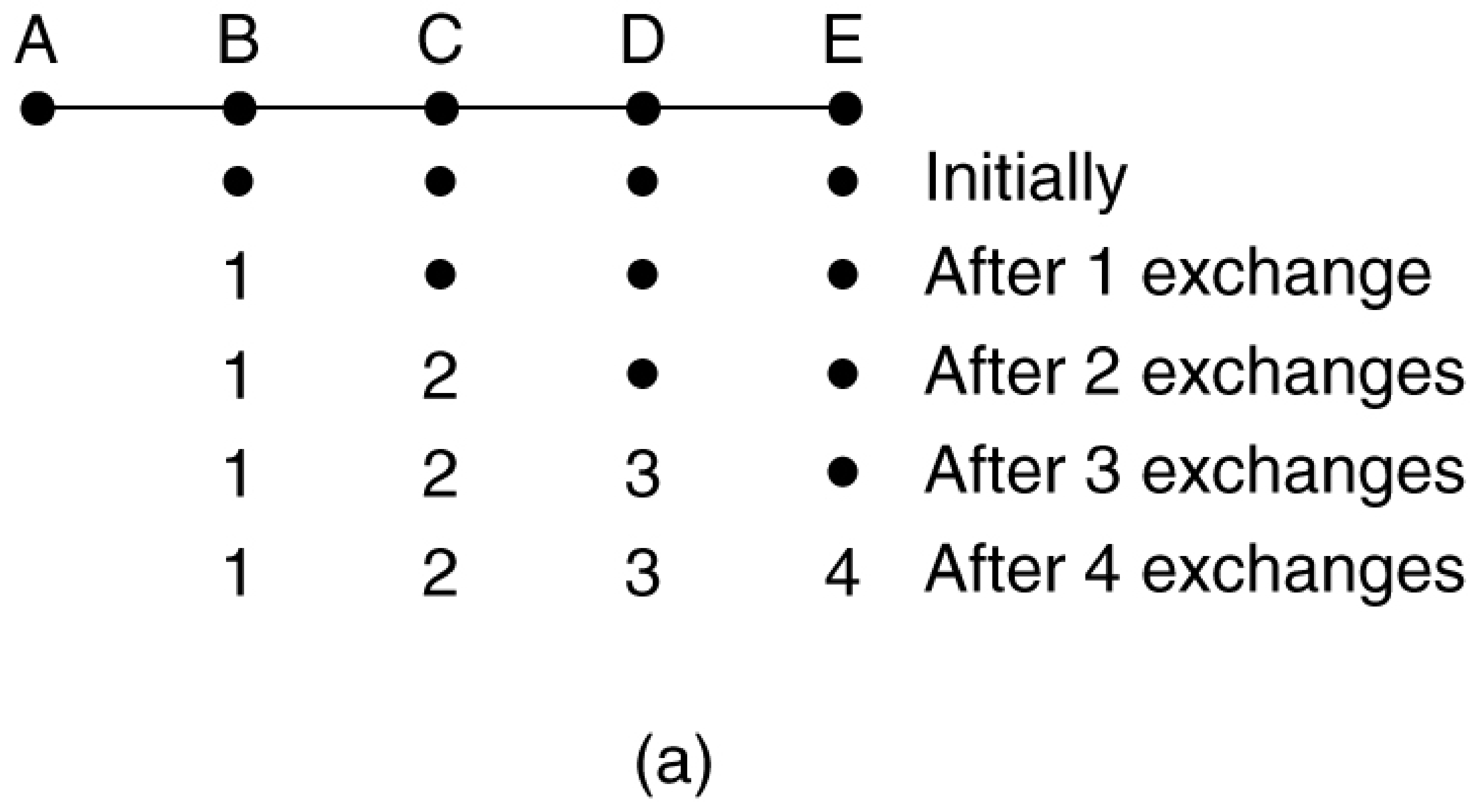
Distance Vector Routing Algorithm

- Drawback
  - Distance vector routing finds the correct shortest path but it may be slowly.

The Count-to-Infinity Problem

- The distance vector routing algorithm has a serious problem when a link goes down (or) comes up otherwise it works well when all the nodes exchange their updated distance vectors. This is called **Count-to-infinity problem**.
- **Example**: Consider the Five-node (linear) subnet, where the delay metric is the number of hops.

The Count-to-Infinity Problem



The count-to-infinity problem

The Count-to-Infinity Problem

- Initially, A is down and all other routers know this and made as infinity.
- When A comes up, at first exchange, B learns that its left neighbor has ‘0’ delay to A. So B makes an entry in routing table that A is one hop away to left.

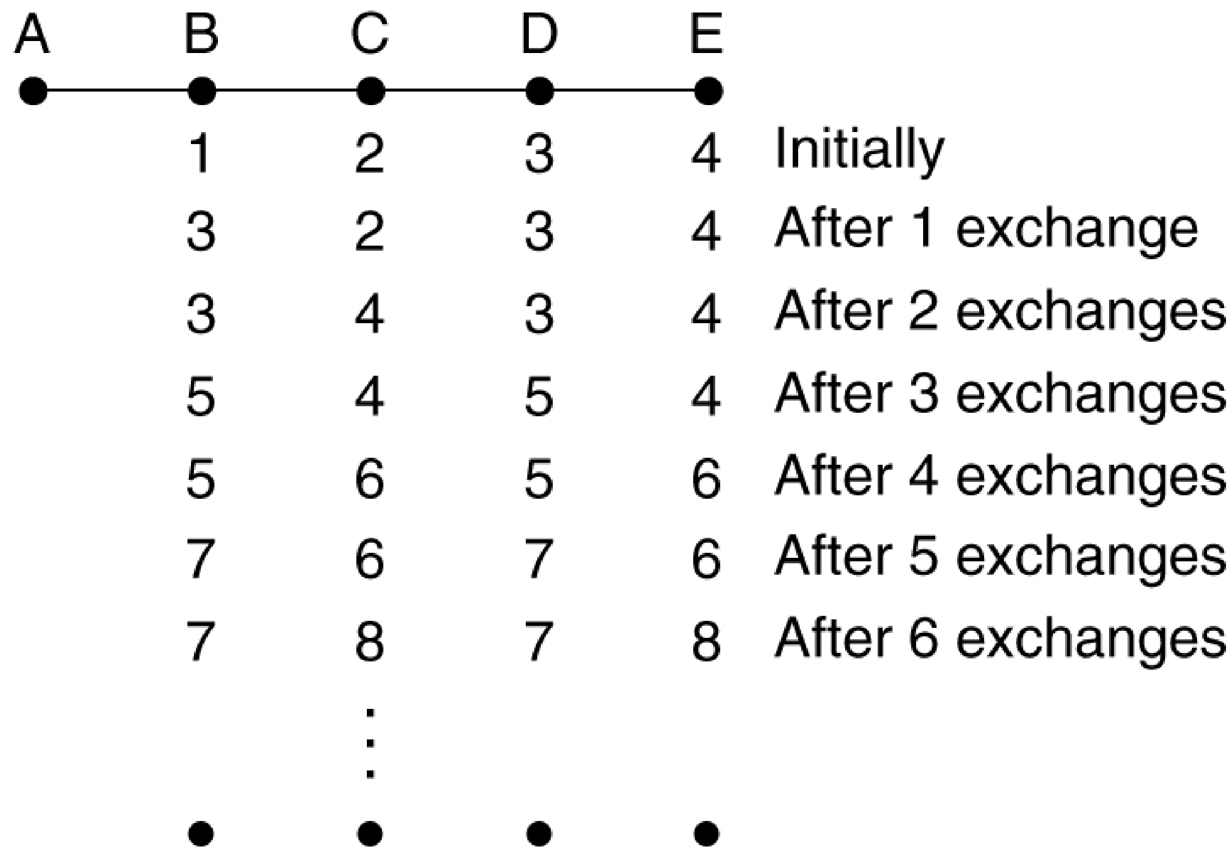
The Count-to-Infinity Problem

- All other routers still think that A is down on the second exchange, C learns that B has a path of length 1 to A, so it updates its routing table to indicate a path of length 2. similarly process is done for all other nodes present in the subnet.

The Count-to-Infinity Problem

- Now let us consider the other situation in fig(b), in which all the lines and routers are initially up. Routers B, C, D and E have distances to A of 1, 2, 3 and 4. If suddenly router A goes down, the line A and B is cut.

The Count-to-Infinity Problem



(b)

The count-to-infinity problem

The Count-to-Infinity Problem

- At the first packet exchange, B does not hear anything from A. Then router C says: Do not worry; I have a path to A of length 2. So, B thinks it can reach A via C, with a path length of 3. Similar process is done for throughout the subnet. That may lead to infinity. So, this problem known as Count-to-Infinity problem.

The Count-to-Infinity Problem

- To overcome the drawback of count-to-infinity problem we use split-horizon algorithm.
- In split-horizon algorithm, a router never advertises the distance to router X to its neighbor N. In above example, when C sends packets to A via B, C tells simply like "I don't know". When the router A goes down, at first packet exchange B notices that A is down and C also telling an infinite distance to A. Since B cannot reach to A via its neighbors it sets distance to A is infinity.

The Count-to-Infinity Problem

- On the next exchange C realizes that A is not reachable since neighbor of its neighbors can reach A. Therefore C also sets its distance to A is infinity. Thus the count-to-infinity problem is solved.

Link State Routing

- Link state algorithm is a dynamic routing algorithm that takes into account the complete topology, all delays and bandwidth when choosing routes.
- In this algorithm, each node sends only the state of the directly connected neighbors. But this information is sent to every node in the subnet.

Link State Routing

In this algorithm, each router must do the following:

1. Discover its neighbors, learn their network address.
2. Measure the delay or cost to each of its neighbors.
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 other router.

- The link state algorithm uses the “Dijkstra’s Algorithm” to find the shortest path.

Learning about the Neighbors

- When a router joins the network, it first learns about it neighbors. To achieve this, it sends a special HELLO packet on each outgoing line. When the packet arrive at the receiver, each receiver reply to it by telling its identity.

Learning about the Neighbors

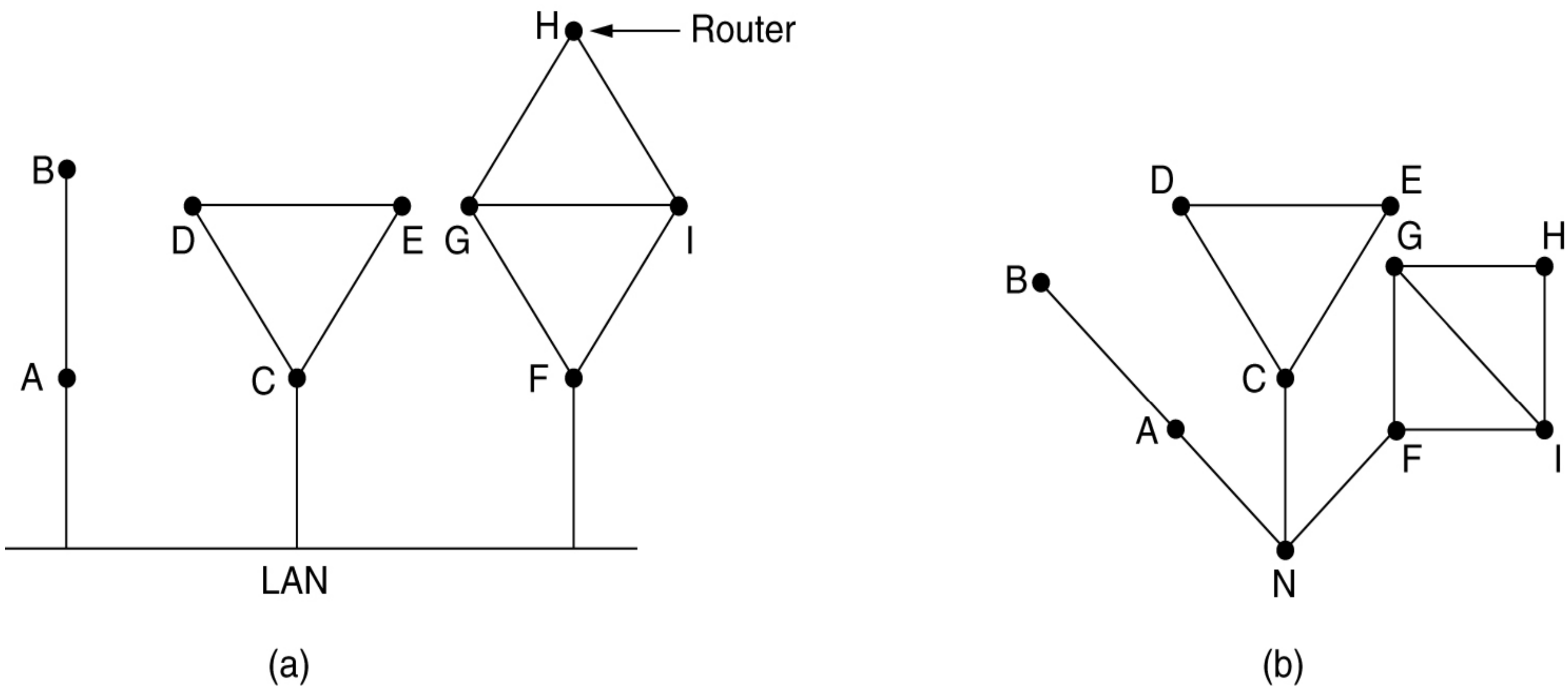
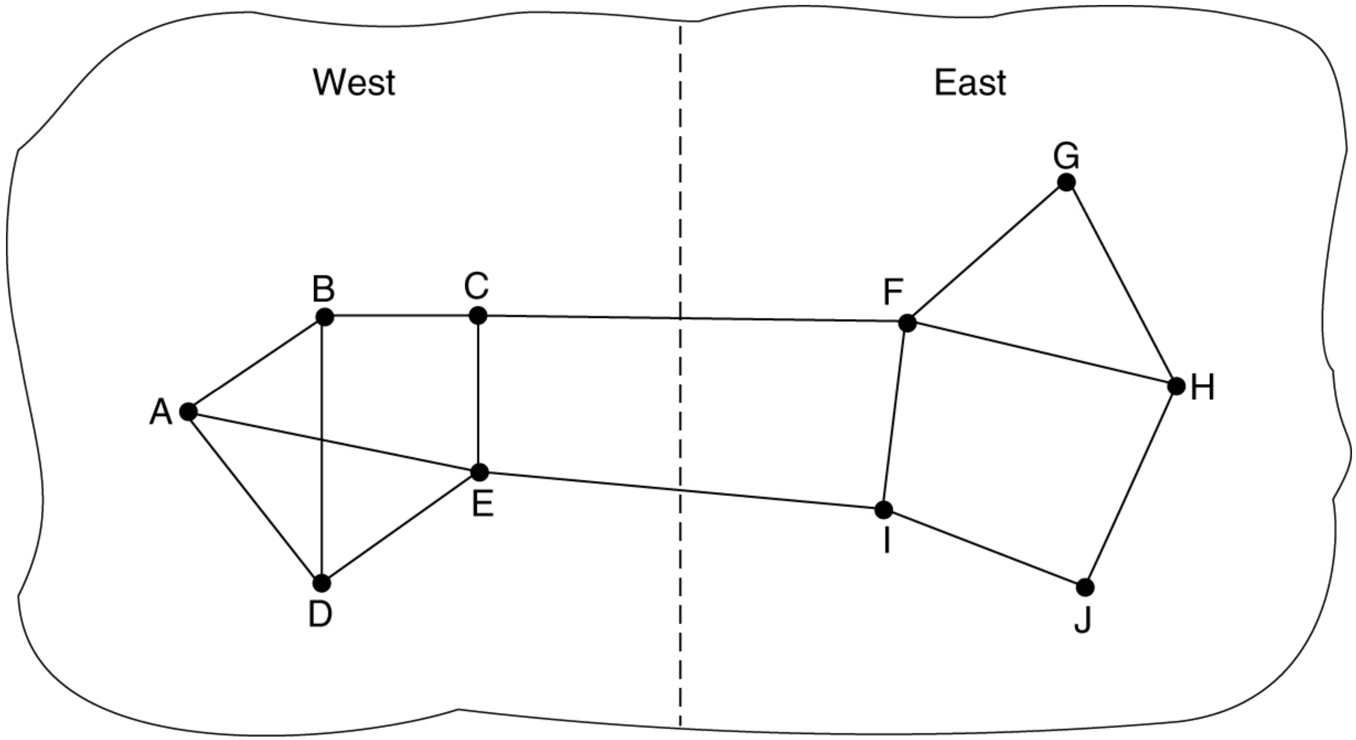


Fig (a) Nine routers and a LAN. Fig(b) A graph model of (a).

Measuring Line Cost

- Next the router must estimate the delay to reach each of its neighbors.
- To know this delay the router send ECHO packet on each point-to-point line.
- The router on other end sends back it immediately.
- The router calculate the delay by dividing the Round-Trip Time (The time a packet take to reach to destination router and come back.)

Measuring Line Cost



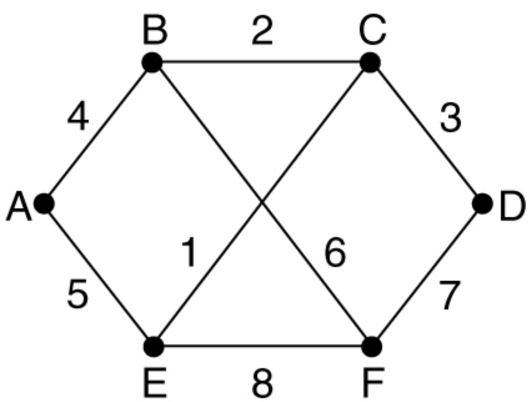
A subnet in which the East and West parts are connected by two lines.

Building Link State Packets

- Once the router gets the information about its neighbors it calculates its delay to them, the next step for each router is to construct a packet including all the information.
- The packets includes the identity of the sender, the sequence number and age, followed by the list of each of its neighbors with its delay to them.

Building Link State Packets

- Example: Consider the subnet shown in Fig(a). The link state packets that each router construct for this subnet in Fig(b).



(a)

Link State Packets

A		B		C		D		E		F	
Seq.		Seq.		Seq.		Seq.		Seq.		Seq.	
Age		Age		Age		Age		Age		Age	
B	4	A	4	B	2	C	3	A	5	B	6
E	5	C	2	D	3	F	7	C	1	D	7
		F	6	E	1			F	8	E	8

(b)

(a) A subnet. (b) The link state packets for this subnet.

Distributing the Link State Packets

- The next step after building the link state packets is to distribute them across the network.
- Flooding is used as basic algorithm for distributing links state packet. To avoid flooding same packet, each new packet is given a sequence number. When a packet arrives at router for flooding then it checks whether this packet is already visit (or) not. If the duplicate packets arrives then it is discarded.
- If a packet with lower sequence number arrives after seeing a packet with highest number then it is rejected.

Distributing the Link State Packets

- Each router uses a data structure with the field source router, packets sequence number and age, send flags and acknowledgment flags and the data.
- Once the age becomes zero, the router discards the information from it.
- The sends flags specify the line on which to send the incoming packet and the acknowledgment flags specify the line on which an acknowledgment is to be sent.

Distributing the Link State Packets

Source	Seq.	Age	Send flags A	Send flags C	Send flags F	ACK flags A	ACK flags C	ACK flags 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	

The packet buffer for router B in the previous slide (Fig. 5-13).

Compute the new route

- The link state algorithm uses the Dijkstra’s algorithm to compute the shortest path to all possible destinations.

Link State Routing

Disadvantage

1. Large memory is required at each node to store the information.
2. More computation time is needed to calculate end-to-end routes.
3. It creates problems if the router does the routing / calculation wrong.

Uses

1. Link state routing is widely used in actual networks

Solutions to link state routing

- The solution to the scaling problem is to do the routing hierarchically.
- Hierarchically routing divides the routers into regions, regions into clusters, the clusters into zones.
- By using this routing reduces the memory required at each node.

Hierarchical Routing

- Hierarchical routing is an algorithm for routing packets hierarchically. This routing is used due to following reasons,

1. Routers need more memory space to store routing table which increased with increase in the network size.
2. More CPU time is needed to scan each routing table
3. More bandwidth is required to send scanning reports
4. It is infeasible for each router to have the knowledge of every other router in large network.

- Therefore routing is done hierarchically as it is used.

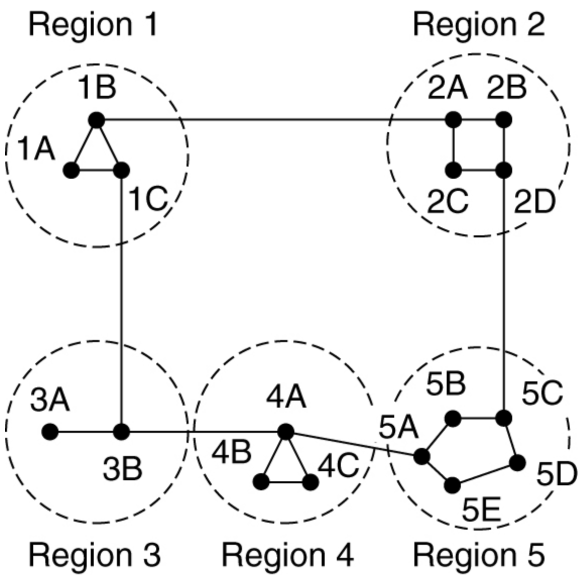
Hierarchical Routing

- Hierarchical routing algorithm divides the routers into regions with a few number of routers with a few number of routers per regions. With this algorithm, each router maintains a routing table that gives detail information about how to route packets to destinations within its own region but it does not tell how to route the same to destinations within other regions.

Hierarchical Routing

- Thus the use of regions reduces the space required by each router and the CPU time needed to scan the routing tables.
- For huge networks, the division of routers into regions is a two-level hierarchy which is insufficient.
- Consider the example of routing in a two-level hierarchy with five regions.

Hierarchical Routing



(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)

Hierarchical routing.

Hierarchical Routing

- In the full routing table, the router 1A has 17 entries as shown in fig(b). When the routing is done hierarchical there are only 7 entries for router 1A.
- These are entries for local routers but all other regions have been present within a single router as shown in fig(c).
- This routing increases saving the table space. So all traffic for region 2 goes via the 1B-2A line.

Hierarchical Routing

- Hierarchical routing has reduced the table from 17 to 7 entries. When the single network becomes very huge, now we can use number of levels of hierarchy.
- Three level hierarchy means group the regions into clusters, the clusters into zones, the zones into group and so on.
- For huge networks, this two-level hierarchy is insufficient.

Hierarchical Routing

- Consider a subnet with **720 routers**. If there is **no hierarchy**, each router needs 720 routing table entries. If there is **two-level hierarchy** used, then the subnet is partitioned into 24 regions of 30 routers each, each router needs 30 local entries plus 23 remote entries for a total of **53 entries**.
- Consider a subnet consists of **720 routers**, if **three-level hierarchy used**, then with 8 clusters, each containing 9 regions of 10 routers, each router needs 10 entries for local routers, 8 entries for routing to other regions within its own clusters, and 7 entries for distant clusters for a total of **25 entries.**

Hierarchical Routing

- **Advantage**
  - The use of regions reduces the memory space and scan time
  - Bandwidth is reduced

UNIT 2 PART 3

Congestion Control Algorithms: General Principles of Congestion Control, Congestion Prevention Policies, Congestion Control in Virtual-Circuit Subnets, Congestion Control in Datagram Subnets, Load Shedding, Jitter Control.

Congestion

- When too many packets are present in the subnet and as traffic increases then the network performance degrades and no packets are delivered to destination. This situation is called a **congestion**.

Congestion Control

- It is a process of maintaining the number of packets in a network at certain level. It make sure that subnet is able to carry the offered traffic.

Effects on congestion

- When too many packets are present in the subnet, congestion occur and network performance degrades as shown in below figure. The number of packets delivered are proportional to the number of packets sent.

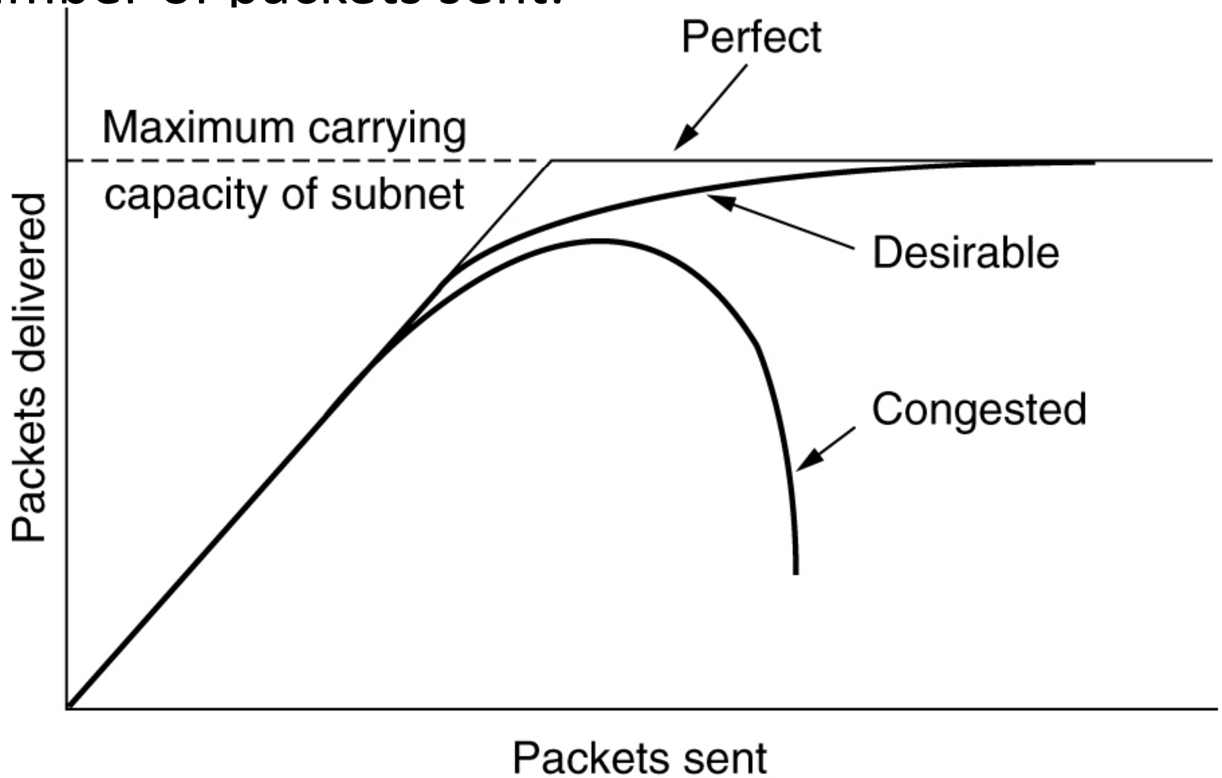


Fig: Effects on congestion

Causes (or) Reasons for congestions

- If there is inefficient memory, to hold all of them, packets will be lost.
  - Solution is to add additional memory.
- If the CPU processing speed is slow. If CPU is busy in updating the routing tables it ignores the packets stored in the CPU.
- If there is low bandwidth lines.
  - Solution is upgrading the lines, but not changing the processors.
- If the traffic on the network is very high.

General Principles of Congestion Control

- Congestion control is concerned about controlling traffic entry into a network.
- Congestion control strategy is divided into two types. They are

1. Open loop congestion control
2. Closed loop congestion control

Open loop congestion control

- It is used for preventing congestion from happening to occur on first place. It is a static control.
- Various open loop congestion control techniques are as follows.

a. Retransmission policy
b. Window policy
c. Acknowledgement policy
d. Discarding policy
e. Admission policy

Retransmission Policy

- If sender feels that a sent packet is lost, the packets needs to be retransmitted.
- Retransmission may increase congestion in the network.
- By using effective retransmission policy and timer, congestion can be controlled.

Window policy

- By specifying the receiver’s window size before the transmission starts, congestion can be avoided as sender transmits the packet according to the size of receiver’s window.

Acknowledgment policy

- This policy make by receivers may also effect congestion.
- If the receiver doesn’t acknowledge every packet it receives, it may slow down the sender and congestion occurs.
- A receiver may send an acknowledgment only if it has a packet to be sent, then the congestion can be controlled.

Discarding policy

- A good discarding policy by the routers may prevent congestion, at the same time it doesn’t affect the transmission packets.

Admission policy

- In this policy, a new virtual connection request is not accepted if that request leads to congestion.

Drawbacks

- The drawback of open-loop congestion control is that once the system has started running in between, connection can not be adjusted.

Closed loop congestion control

- It is used to remove the congestion, after it occurred. These congestion control strategies are based on feedback mechanisms. This approach has three parts when applied to congestion control:

1. Monitor the system to detect when and where congestion occurs.
2. Pass this information to places where action can be taken.
3. Adjust system operation to correct the problem.

Closed loop congestion control

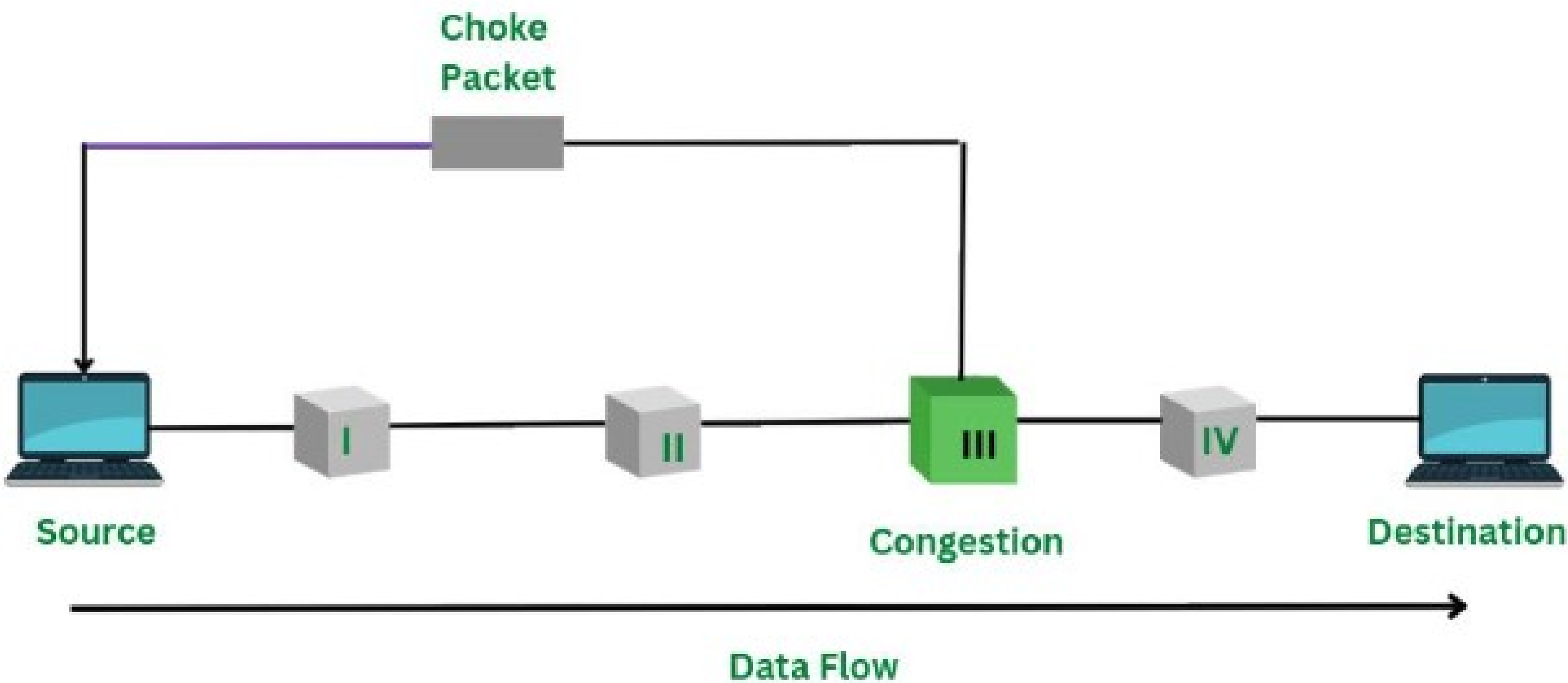
- Different kinds of closed loop congestion control techniques as follows:

1. Choke packets
2. Implicit signaling
3. Explicit signaling

Choke packets

- A choke packet is a packet sent by a node to the source to inform it of congestion.
- In this method, the warning is from the router about congestion occur, to the source stations directly.

Choke Packets Mechanism



Implicit signaling

- In this, the source automatically learn about congestion by observing the delay in packets acknowledgement, time-out determination. It thinks that congestion occur in network.

Explicit signaling

- In explicit signaling, packets are sent back from the point of congestion to warn the source.
- In this, the information about congestion is send along with the packet carrying data which is different from choke packets. It can occur either in **backward signaling** or **forward signaling**.

Backward signaling

- A bit can be set in a packet moves in the opposite direction to the congestion. This bit tells the source that there is a congestion.

Forward signaling

- A bit can be set in a packet moving in the direction of congestion. This bit can warn the destination that there is a congestion.
- Mainly congestion means load is greater than the resources that they can handle.
- Two solutions can be applied to control the congestion are increase the resources and decrease the load.
- Increase the resources
  - The subnet may use dial-up telephone line to temporarily increase the bandwidth
  - Split the traffic over multiple lines instead of using the same best line
  - It is not possible to increase the resources. Because it is a matter of cost.

Summary

- In open loop by considering congestion problems it should design the subnet statically.
- In closed-loop after arising problems it was given as feed back

Congestion prevention policies

- There are some congestion prevention policies for different layers like Datalink, Network, Transport layer as shown in below table.

Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Policies that affect congestion

Out-of-order caching policy

- If receiver continuously discard all Out-of-order packets, these packets will have to be transmitted again later, creating extra load by using protocol like Go-Back-N and selective repeat.
- With respect to congestion control, selective repeat is clearly better than go back n.

Flow control policy

- When sender sending data at very high rate and receivers receiving at very low rate then automatically congestion will occur. So we have to use flow control mechanism.
- Solution is to use small window size.

Routing Algorithm

- A good routing algorithm can help avoid congestion by spreading the traffic over all the lines, whereas a bad one can send too much traffic over already congested lines.

Packet lifetime management

- It deals with how long a packet may live before being discarded.
- If it is too long, lost packets may clog up the works for a long time, but if it is too short, packets may sometimes time out before reaching their destination, thus inducing retransmissions.

Packet queuing and service policy

- It relates to whether routers have one queue per input line, one queue per output line or both.
- It relates to order, packets are processed. (Priority, Round Robin, FIFO)

Packet Discard Policy

- When there is no sufficient space in the queues, discard policy is used to decide which packet has to be discarded.
- If the time out is too short, extra packets will be sent unnecessarily.
- If it is too long, congestion will be reduced but the response time will suffer whenever a packet is lost.

Organizing the subnet

- There are two ways for organizing the subnet. They are

a) Virtual-circuit subnet

b) Datagram subnet

Virtual circuit subnet

- Virtual circuit subnet is a connection-oriented service, a path is established from source to destination before the transmitting data packet.
- This path is called virtual circuit and corresponding subnet is called virtual circuit subnet.

Advantages and Disadvantages

Advantages

- Sequence of packets is guaranteed
- Header of shorter length can be used that contains virtual circuit number

Disadvantages

- For each ongoing connection, tables are needed to be stored at each router
- Router failures cannot be recovered

Datagram Subnet

- Datagrams are connection-less and the subnet corresponding to datagram is called **datagram subnet**.
- Router maintains the table which specify the destination and output link that is used to send datagram to that destination.

Advantages and Disadvantages

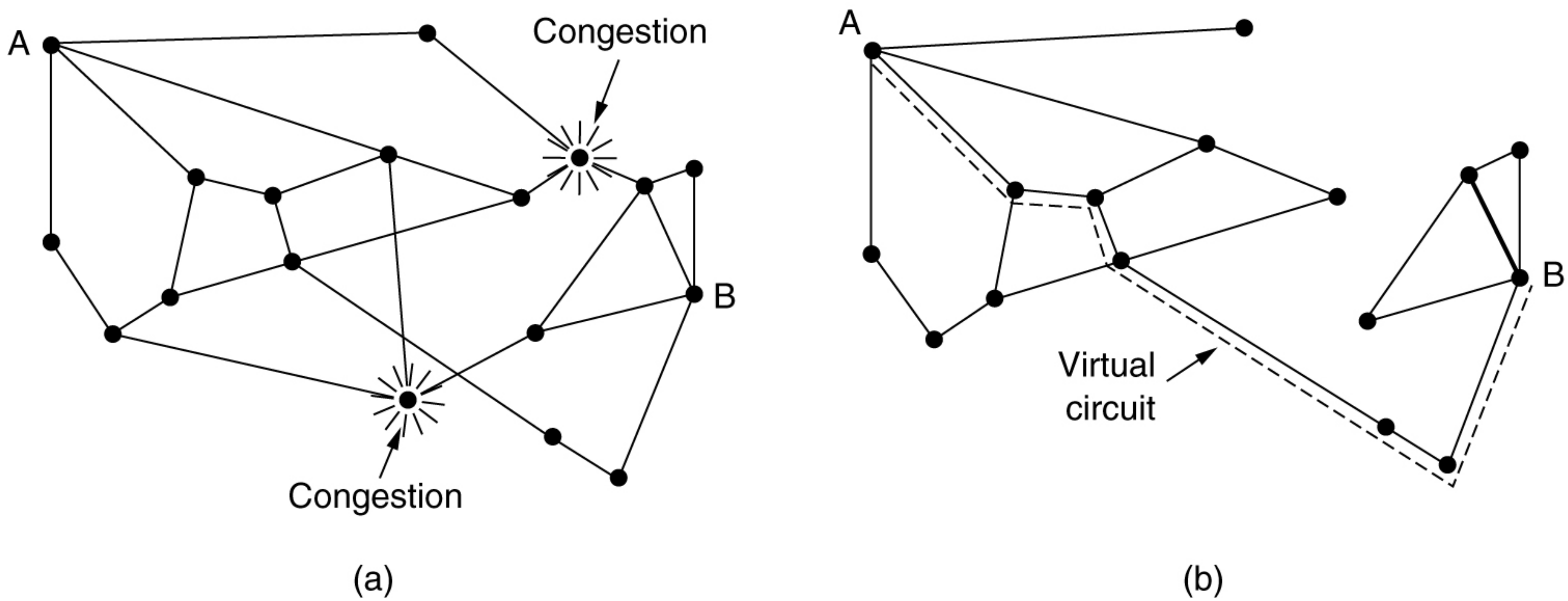
- **Advantages**
  - Robust to deal with router fails
  - It has capability to deal with congestion.
- **Disadvantage**
  - Sequence of packet is not guaranteed at the destination.
  - Quality of service cannot be provided to real-time application.

Congestion Control in Virtual-Circuit Subnets

- Various mechanisms that are used for controlling congestion in virtual circuit subnet are,
  - **New virtual circuit connection are not established into the subnet once congestion is triggered**. This strategy is known as controlling the admission of the connection called **admission control**. In this congestion already occurred (closed loop)
  - **Allowing virtual circuit connection even after congestion** but using different routes that are not congested.
  - **Reserving the resource in advance**. This strategy results in less utilization of the bandwidth.

Congestion Control in Virtual-Circuit Subnets

For example:- Consider a subnet, in which two routers are congested



(a) A congested subnet. (b) A redrawn subnet, eliminates congestion and a virtual circuit from A to B.

Congestion Control in Virtual-Circuit Subnets

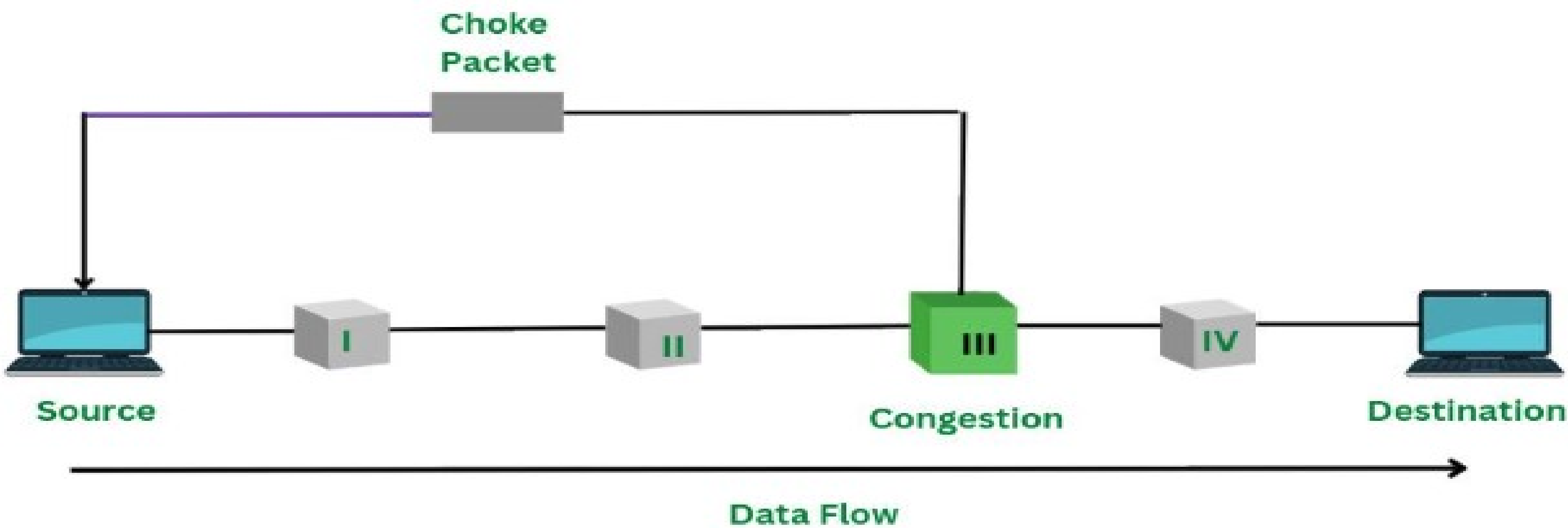
- Suppose that a host attached to router A wants to setup a connection to a host attached to router B. Normally this connection would pass through one of the congested routers.
- To avoid this situation we can redraw the subnet as shown in above fig(b) omitting the congested routers and all of this lines.
- The dashed lines shows a possible route for the virtual circuit that avoids the congested routers.

Disadvantage

- By using this, in the subnet there is a wastage bandwidth (resources).
- If 6 virtual circuits reserves 6 Mbps line. But if they use only 1 Mbps to pass all the traffic (packets) remaining is wasted.

Congestion control in Datagram subnets

- Congestion can be controlled in datagram subnet by using the **choke packet** is sent by a node to the source to inform it of congestion.
- In this method the warning is from the router about congestion occurs to the source station directly.



Congestion control in Datagram subnets

- There are three ways for reducing the traffic. They are

a) The warning bit

b) Choke packets

c) Hop-by hop choke packet

The warning bit

- The old DECNET architecture signaled the warning state by setting a special bit in the packet's header. So does frame relay.
- When the packet arrived at its destination, the transport entity copied the bit into the next acknowledgement sent back to the source.
- The source then cut back on traffic.

Choke packets

- The choke packets will have the effect of stopping (or) slowing down the rate of transmission from source and hence limit the total number of packets in the network.
- Whenever you moves above the threshold, the output line enter a “warning state”. Each newly arriving packet is checked to see if the output line is in warning state. If so, the router sends a choke packet back to the source host giving its destination found in the packet.

Choke packets

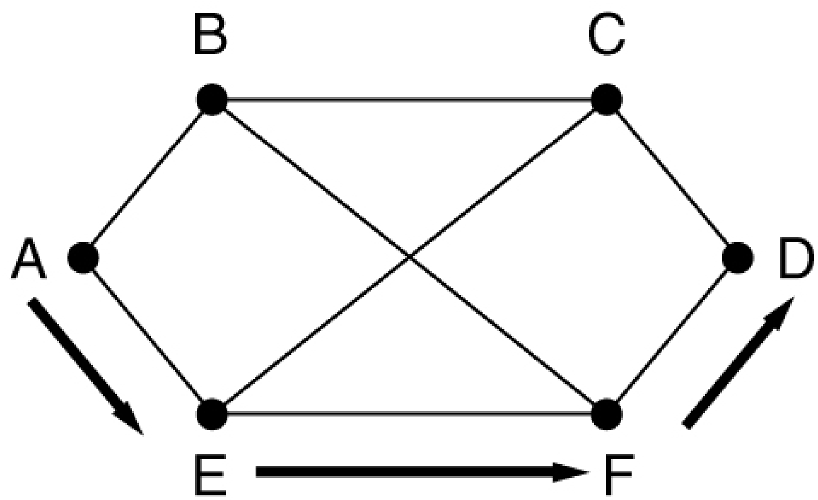
- When the source hosts get the choke packet, it is required to reduce the traffic sent to the specified destination. Typically the first choke packet causes the data rate to be reduced to 50% of its previous rate, the next one causes a reduction to 25%, and so on.
- Increases are done in smaller increments 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 so slow.
- Example: Congestion control using choke packets can be done by two ways.
- In **first type**, the choke packets affects only source as shown in figure(s).

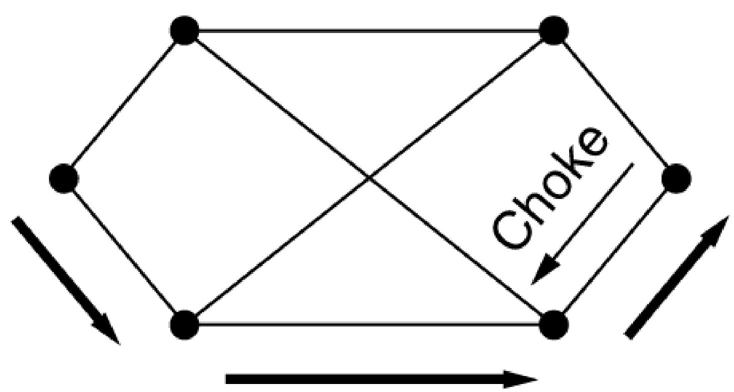
Hop-by-Hop choke packets

- A subnet with 6 nodes A, B, C, D, E, F in below figure, here source node A and destination node is D.



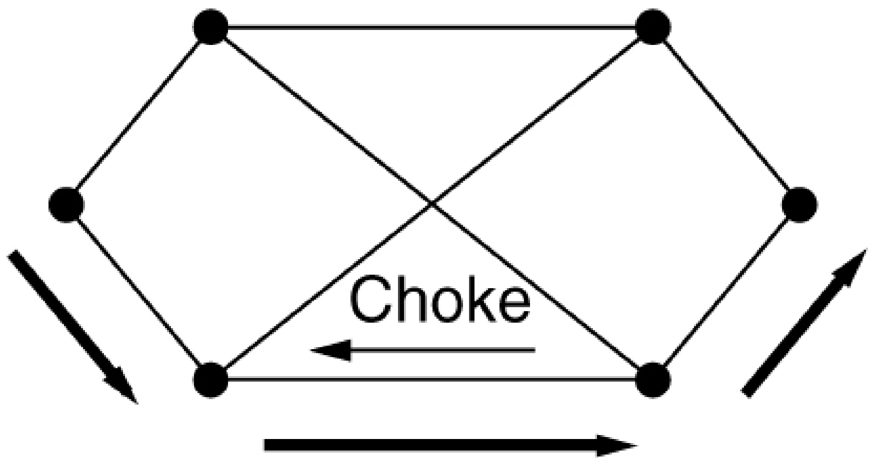
Hop-by-Hop choke packets

- When link utilization increased above its threshold, destination D starts sending choke packets towards source node A.



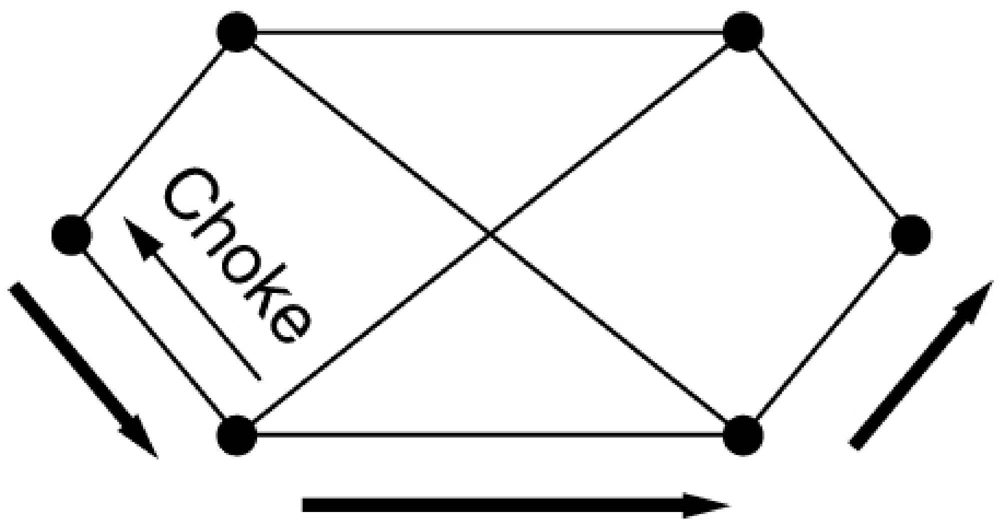
Hop-by-Hop choke packets

- The choke packet travels through source node A.



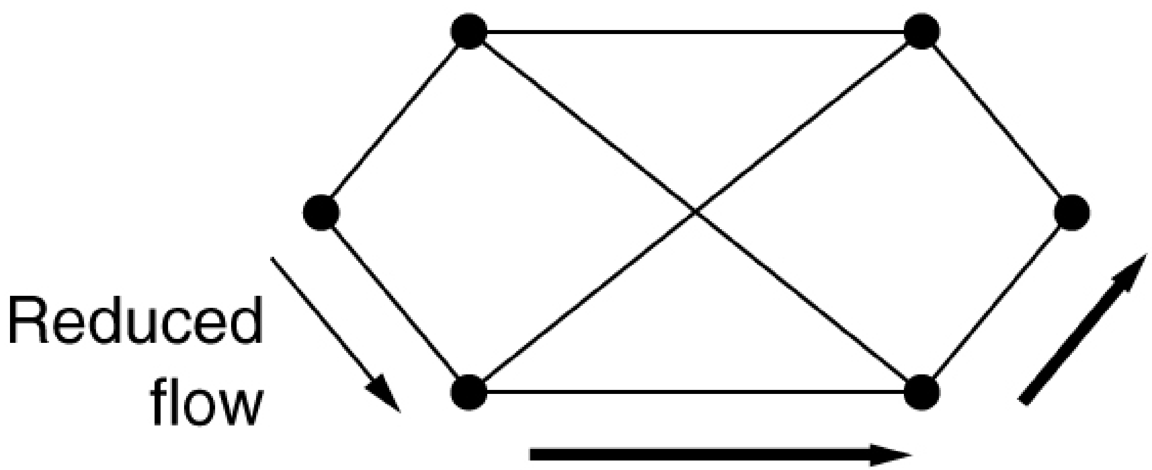
Hop-by-Hop choke packets

- After receiving first choke packet source node A reduces its flow towards destination



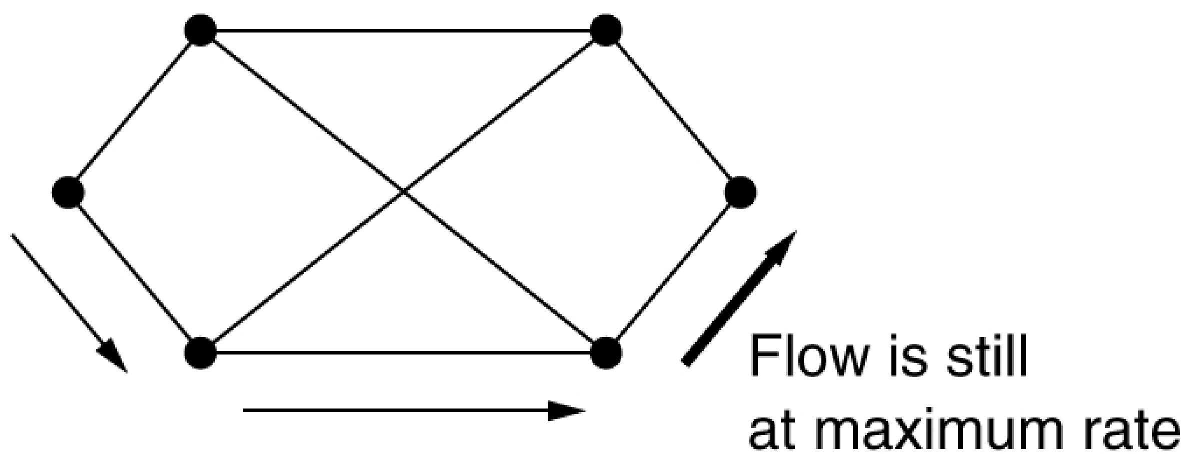
Hop-by-Hop choke packets

- The reduced packet flow follows the same reverse path.



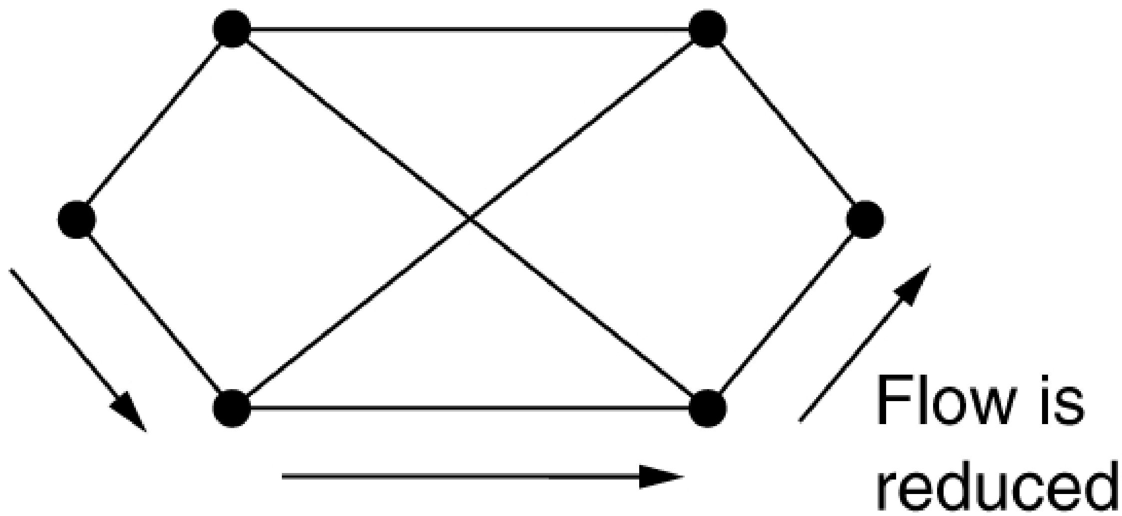
Hop-by-Hop choke packets

- The reduced flow reaches to destination node D.



Hop-by-Hop choke packets

- Flow is reduced



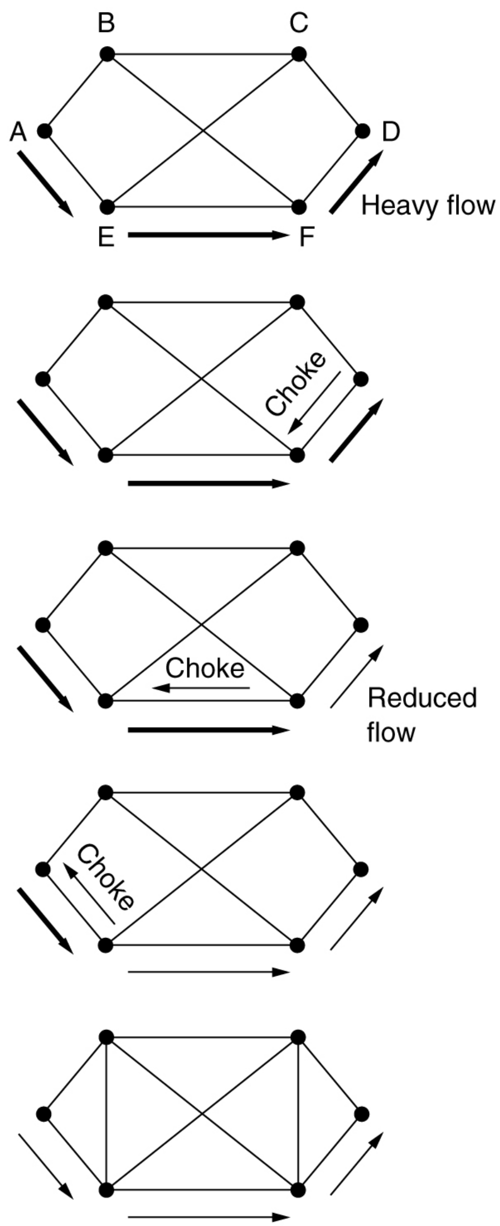
Hop-by-Hop choke packets

- Another way of reducing congestion is that necessary action is taken at each hop to reduce the traffic towards the congested destination after receiving the choke packet. Using this method, congestion is reduced faster.
- Below figure shows that a choke packet that affects each hop it passes through.

Hop-by-Hop choke packets

- Example: The choke packets that affects on each hop it passes through. For the same subnet having nodes A, B, C, D, E, F source node A and destination node D.

Hop-by-Hop choke packets



After reaching choke packets to source node A, the traffic flow between node A, node E and hence the upto destination node D is reduced.

Load Shedding

- It is applied when none of the technique solve the congestion.
- Load shedding is a **discarding policy** in which packets can be discarded if the load of packets are not handled by the router.
- It is analogous to distribution of electricity to certain areas which is ON and OFF in some another area, so that the generated power can be distributed properly.

Load Shedding

- When a packet arrives, which packet will be discarded is basically depends on type of application and their importance.
- There are two policies
  - Wine Policy →Old is better than new
  - Milk Policy →New is better than old

Load Shedding

- While transferring a file, old packets are more valuable and of greater importance than new packets.
- If a new packet is accepted by discarding the old one, it will cause retransmission of packet from an old packet.
- This strategy is called the **wine strategy**.

  Ex: File Transfer Application

Load Shedding

- While transferring **real-time** data such as audio, video, multimedia packet, a new packet is more important than an old packet.
- This strategy is called **milk strategy**.

  Ex: Multimedia Application

Load Shedding

- Another useful way of discarding the packets is by assigning priorities to each packet. Packets with low priority will be discarded first and packets with very high priority will never be discarded.
- Ex - **For ATM networks** priority is marked with using the cell loss priority (CLP) in which value of **"1"** specifies that the cells of higher priority and the value of **"0"** specifies that the cell is lower priority and that it can be **discarded**.

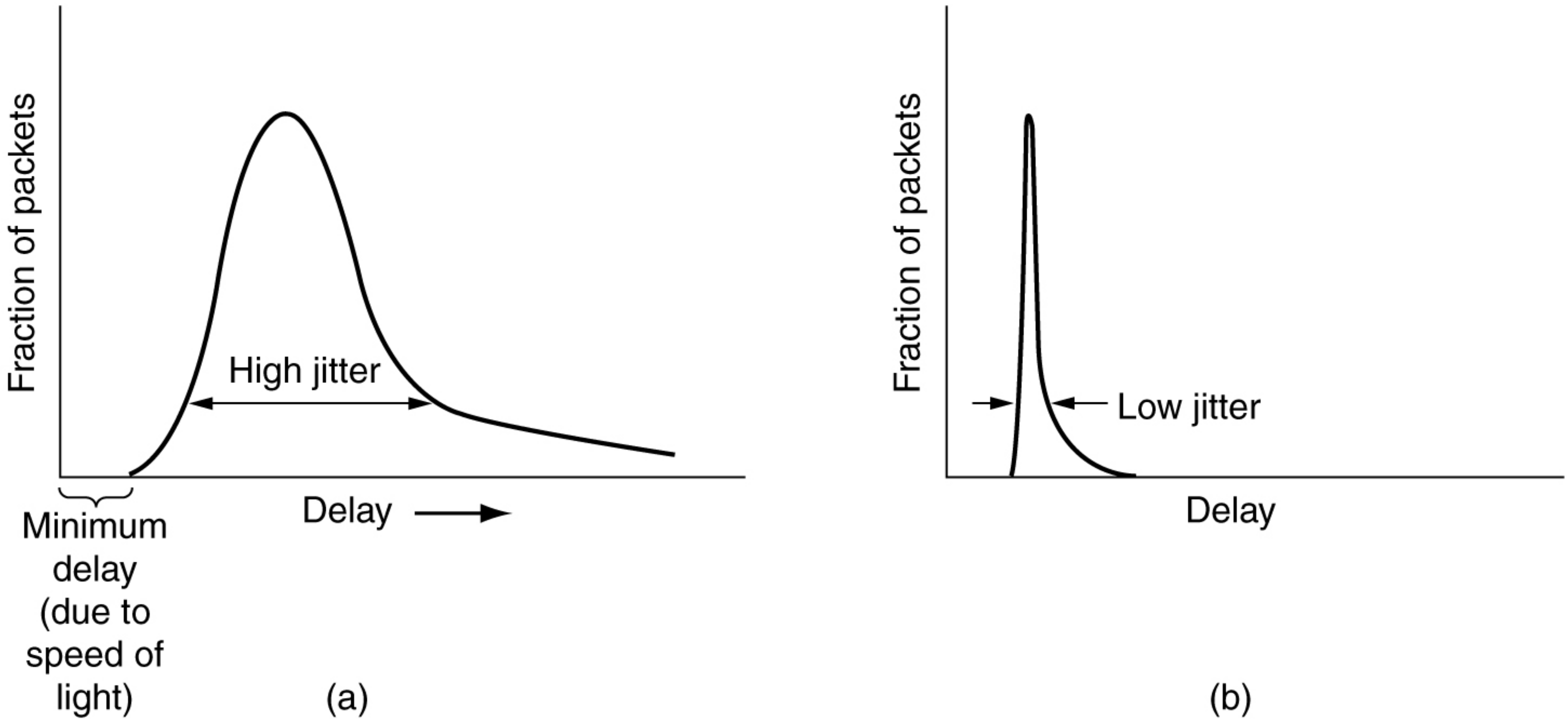
Jitter Control

- The variation (i.e. standard deviation) in the packet arrival times is called **jitter**.
- Jitter control is a mechanism that is used to set the **transmission speed**, so that all the packets are transmitted with the same transmission speed.
- It is basically used to avoid any speed mismatches among the packets that are transmitted, since some packets may be transmitted with higher speed and some with lower speed which results in **uneven quality of data**.

Jitter Control

- High jitter, example having some packets taking 20msec and other taking 30msec to arrive will give an uneven quality to sound (or) movie as shown in below figure.
- In an agreement that 99% of packets be delivered with a delay in range of **24.5msec** to **25.5msec** might be acceptable.

Jitter Control



(a) High jitter. (b) Low jitter.

Jitter Control

- Jitter control does this by calculating the jitter that is, the transmit time expected for each hop along the path. This jitter information is maintained at each intermediate router and is updated with each hop.
- The router checks if a packet that has arrived is behind (or) ahead of its transmit time.
- If the packet is **behind** then its speed is increased. On the other hand if packet is **ahead** of transmit time, then it is held for sometime and transmitted with a low speed.
- In both cases, reducing the amount of jitter.

Jitter Control

- Jitter can be bounded by computing the expected transit time for each hop along the path. When a packet arrives at a router, the router checks to see how much the packet is behind or ahead of its schedule. This information is stored in the packet and updated at each hop.

Jitter Control

- In some applications, such as **video on demand**, jitter can be eliminated by buffering at the receivers and then fetching data for display from the buffer.