UNIT-4 NETWORK LAYER

The 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 and Datagram Networks, Routing Algorithms-The Optimality principle-shortest path, Flooding, Distance vector, Link state, Hierarchical, Congestion Control Algorithms-Leaky bucket & Token bucket.

Internet Working: IP protocols – IP Version 4 protocol IPV4 Header Format, IP Addresses, IP Version 6 – The main IP V6 header, Transition from IPV4 to IP V6.

- The network layer is concerned with getting packets from the source all the way to the destination.
- To achieve its goals, the network layer must know about the topology of the network (i.e., the set of all routers and links)

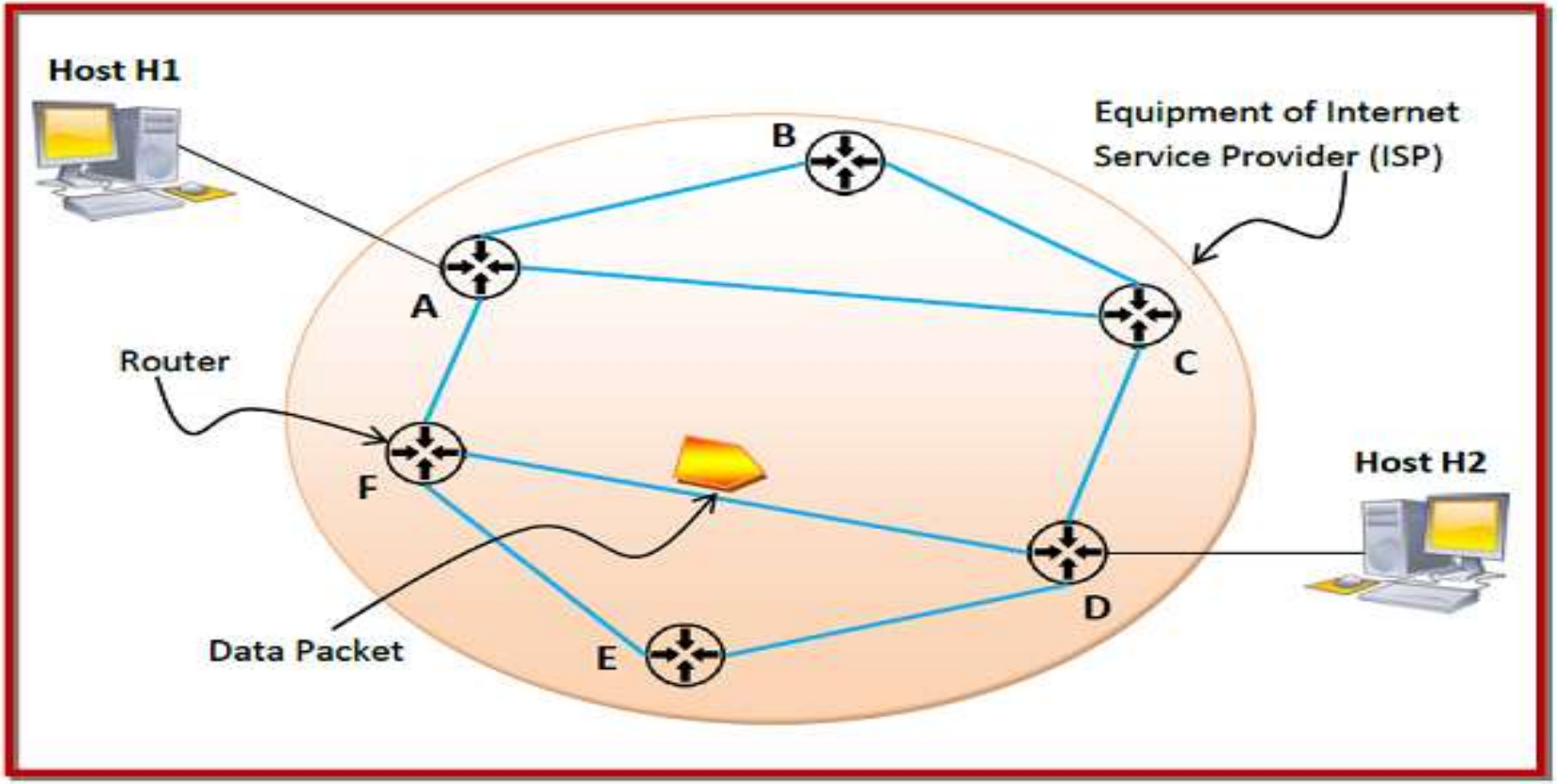
Network Layer Design Issues :

- Store and Forward Packet Switching
- Services Provided to the Transport Layer
- Implementation of Connectionless Service.
- Implementation of Connection-Oriented Service.
- Comparison of Virtual-Circuit and Datagram Networks

Store and Forward Packet Switching:

Working Principle

- The node which has a packet to send, delivers it to the nearest node, i.e. router. The packet is stored in the router until it has fully arrived and its checksum is verified for error detection. Once, this is done, the packet is transmitted to the next router. The same process is continued in each router until the packet reaches its destination.

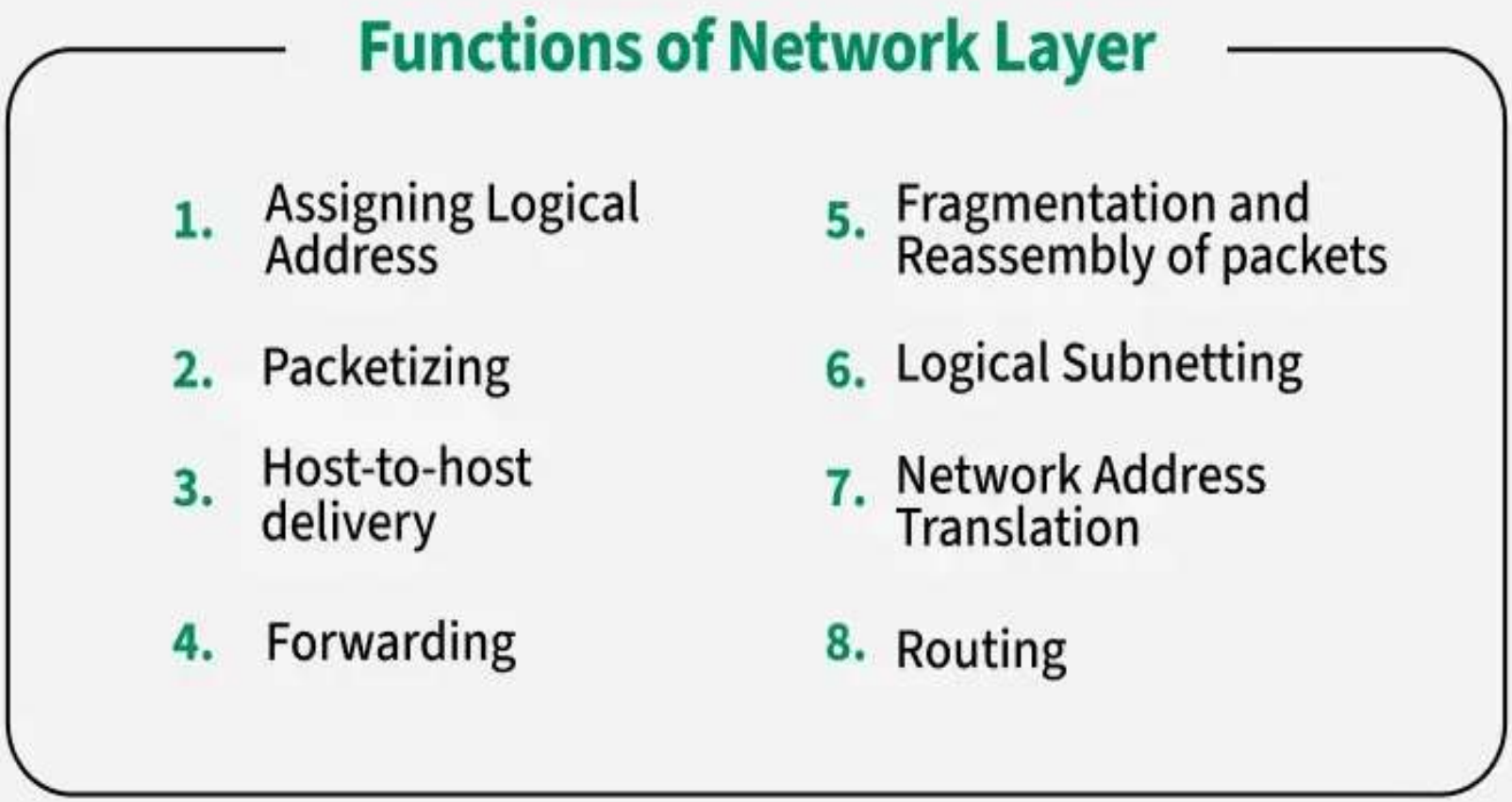


In the above diagram, we can see that the Internet Service Provider (ISP) has six routers (A to F) connected by transmission lines shown in blue lines. There are two hosts, host H1 is connected to router A, while host H2 is connected to router D.

- Suppose that H1 wants to send a data packet to H2. H1 sends the packet to router A. The packet is stored in router A until it has arrived fully. Router A verifies the checksum using CRC (cyclic redundancy check) code. If there is a CRC error, the packet is discarded, otherwise it is transmitted to the next hop, here router F. The same process is followed by router F which then transmits the packet to router D. Finally router D delivers the packet to host H2.

Services Provided to the Transport Layer:

- The **services** which are offered by the network layer are as follows:

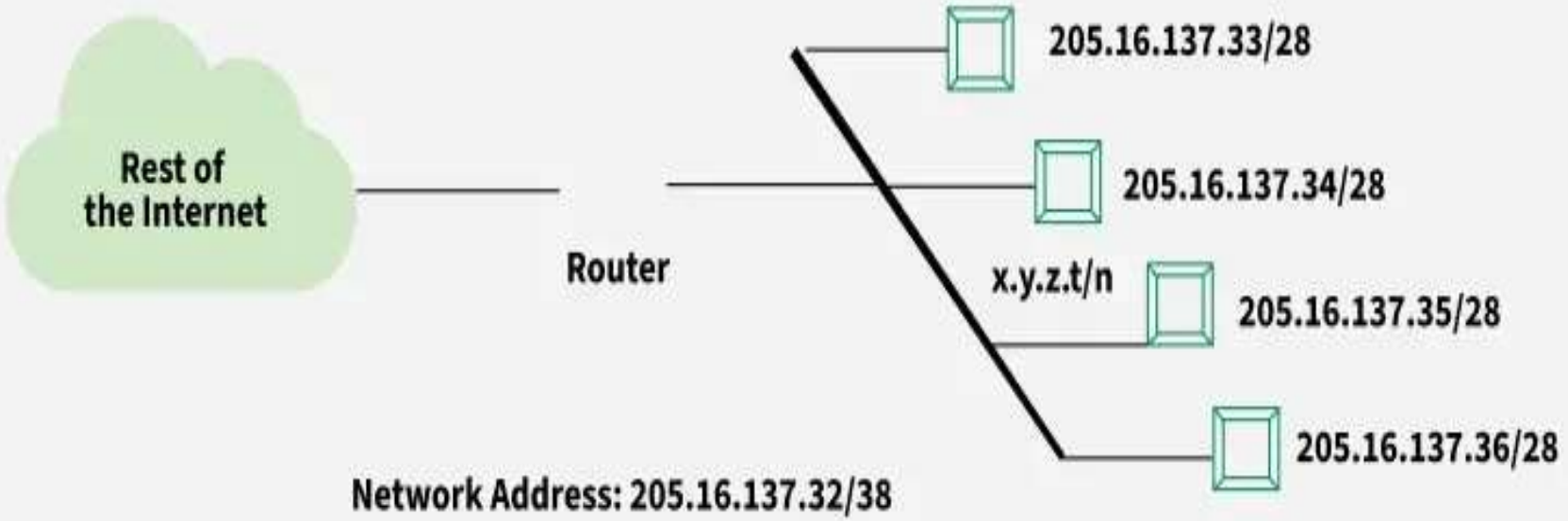


1. Assigning Logical Address

- Logical addressing is the process of assigning unique IP addresses (IPv4 or IPv6) to devices within a network. Unlike physical addresses (MAC addresses), logical addresses can change based on network configurations. These addresses are hierarchical and help identify both the network and the device within that network. Logical addressing is important for:
- Enabling communication between devices on different networks.
- Facilitating routing by providing location-based information.

1. Assigning Logical Address

It assigns unique IP addresses to devices for identification across networks.



- Connection-specific DNS Suffix . : srivasavi.local
- Link-local IPv6 Address . . . . . : fe80::3d28:82a8:510:3bef%1
- IPv4 Address. . . . . : 10.10.15.61
- Subnet Mask . . . . . : 255.255.252.0
- Default Gateway . . . . . : fe80::6e72:20ff:fecf:2e5a%10
- fe80::1262:ebff:fe13:22d5%10
- 10.10.12.1

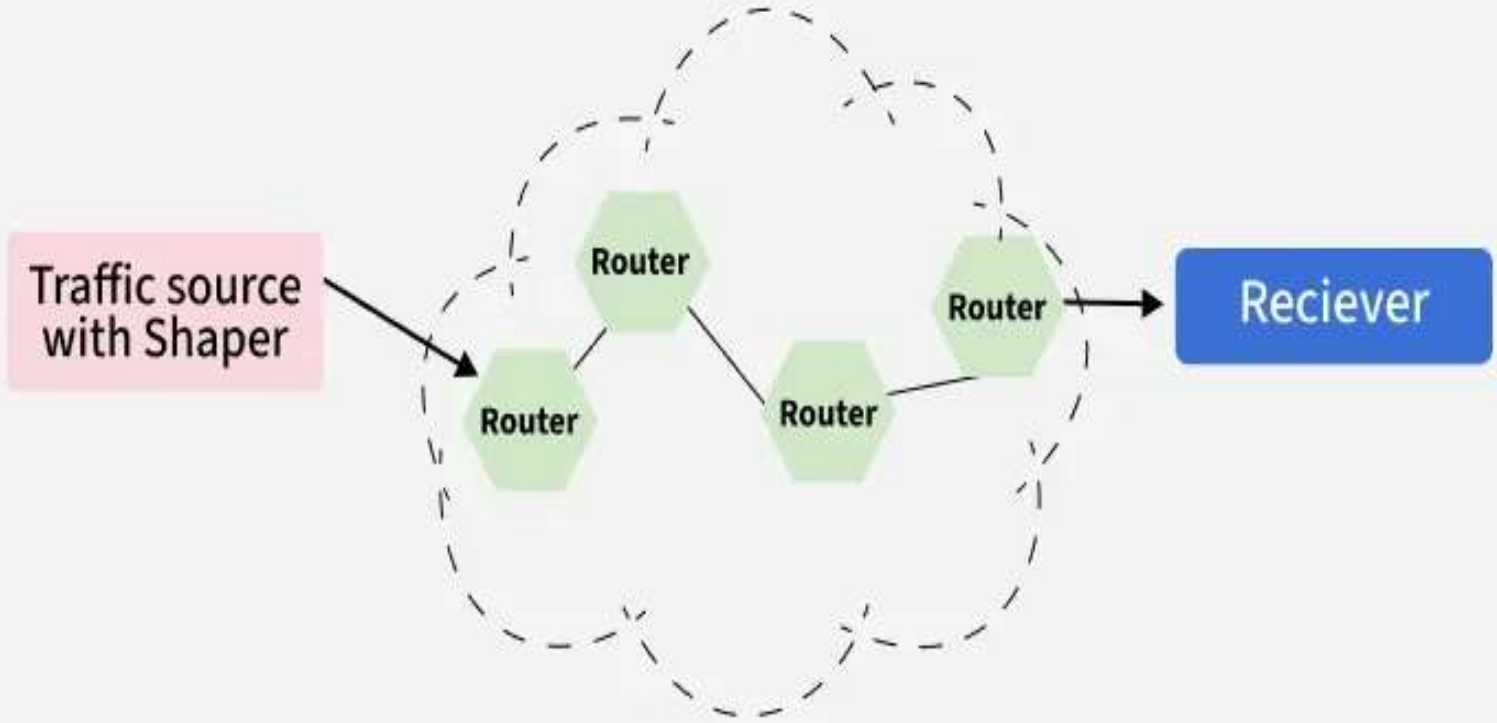
2. Packetizing

The process of encapsulating the data received from the upper layers of the network (also called payload) in a network layer packet at the source and decapsulating the payload from the network layer packet at the destination is known as packetizing.

The source host adds a header that contains the source and destination address and some other relevant information required by the network layer protocol to the payload received from the upper layer protocol and delivers the packet to the data link layer.

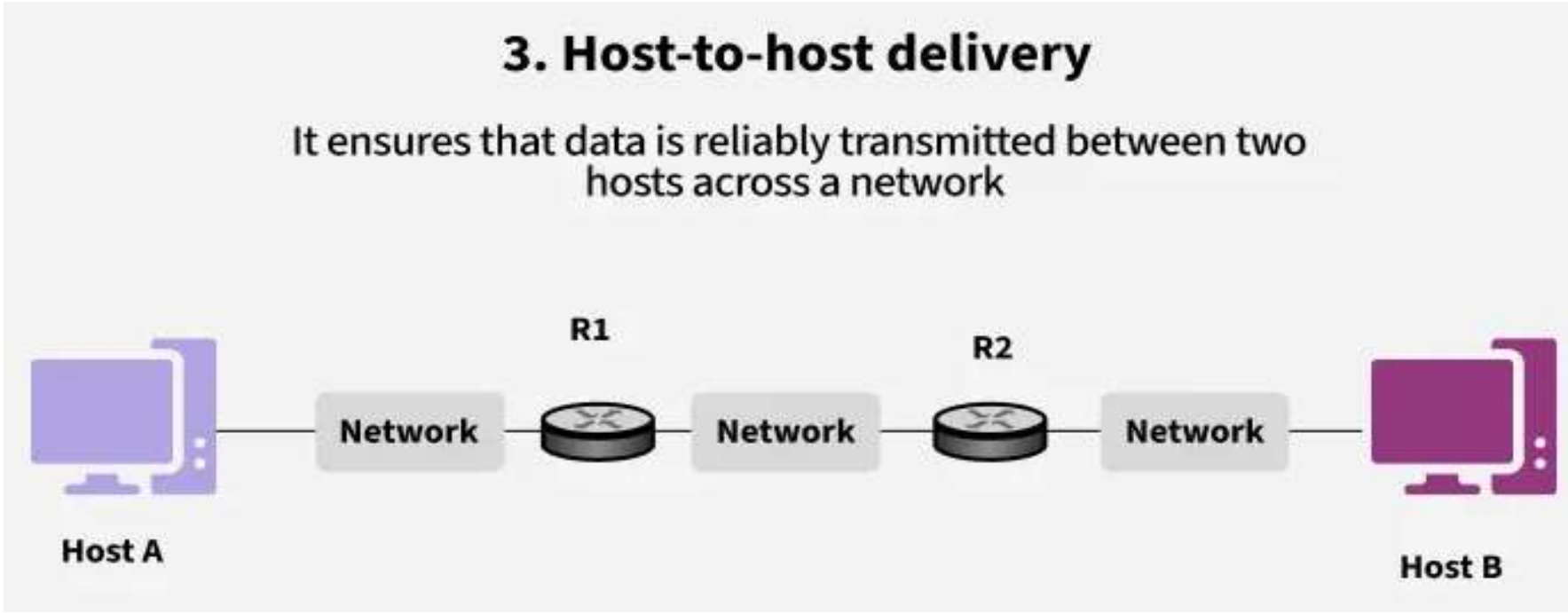
The destination host receives the network layer packet from its data link layer, decapsulates the packet, and delivers the payload to the corresponding upper layer protocol. The routers in the path are not allowed to change either the source or the destination address. The routers in the path are not allowed to decapsulate the packets they receive unless they need to be fragmented.

2. Packetizing



3. Host-to-Host Delivery

- The network layer ensures data is transferred from the source device (host) to the destination device (host) across one or multiple networks. This involves:
- Determining the destination address.
- Ensuring that data is transmitted without duplication or corruption.

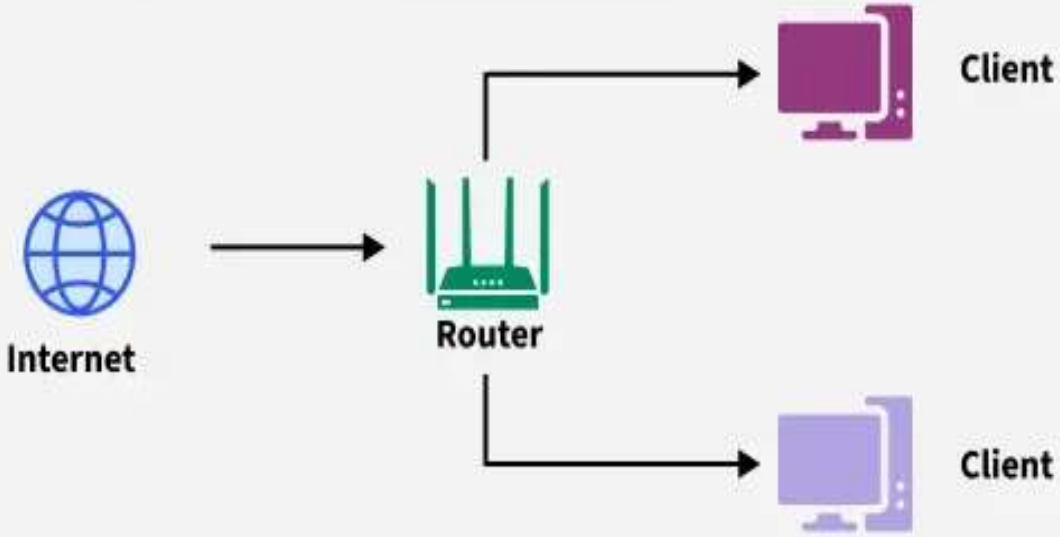


4. Forwarding

- Forwarding is the process of transferring packets between network devices such as routers, which are responsible for directing the packets toward their destination. When a router receives a packet from one of its attached networks, it needs to forward the packet to another attached network (unicast routing) or to some attached networks (in the case of multicast routing).The router uses:
- **Routing tables**: These tables store information about possible paths to different networks.
- **Forwarding decisions**: Based on the destination IP address in the packet header. Forwarding ensures that packets move closer to their destination efficiently.

4.Forwarding

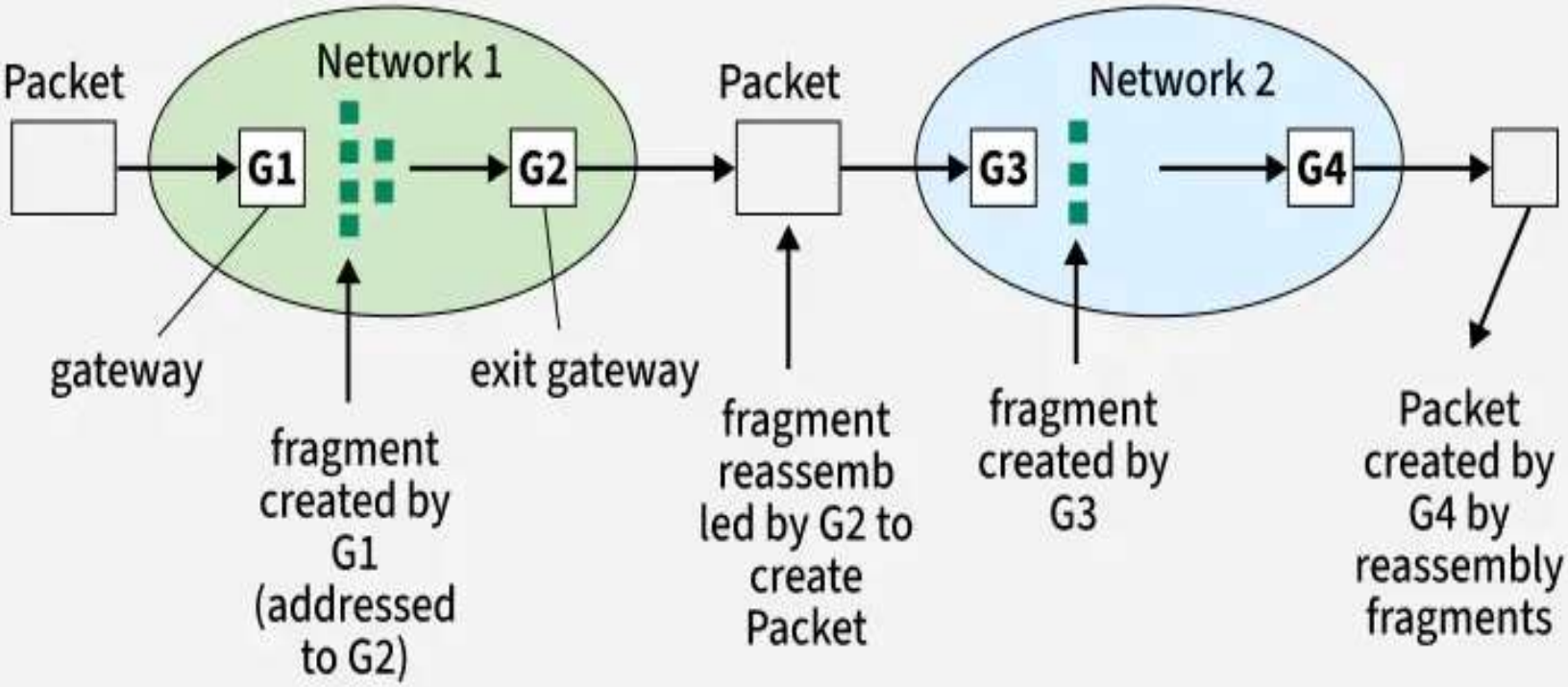
Forwarding moves packets through a router to the correct outgoing interface based on their destination.



5. Fragmentation and Reassembly of Packets

- Some networks have a **maximum transmission unit (MTU)** that defines the largest packet size they can handle. If a packet exceeds the MTU, the network layer f**ragments** the packet into smaller pieces.
- Adds headers to each fragment for identification and sequencing. At the destination, the fragments are **reassembled** into the original packet. This ensures compatibility with networks of varying capabilities without data loss.

5. Fragmentation and Reassembly of packets



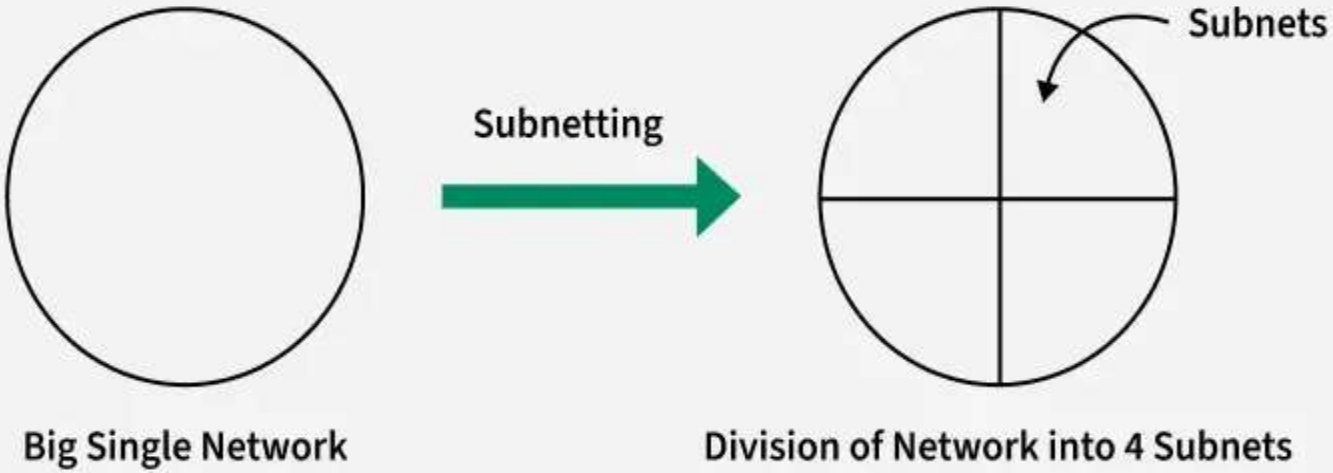
6. Logical Subnetting

- Logical subnetting involves dividing a large IP network into smaller, more manageable sub-networks (subnets).

Subnetting helps:

- Improve network performance by reducing congestion.
- Enhance security by isolating parts of a network.
- Simplify network management and troubleshooting. Subnetting uses **subnet masks** to define the range of IP addresses within each subnet, enabling efficient address allocation and routing.

6. Logical Subnetting

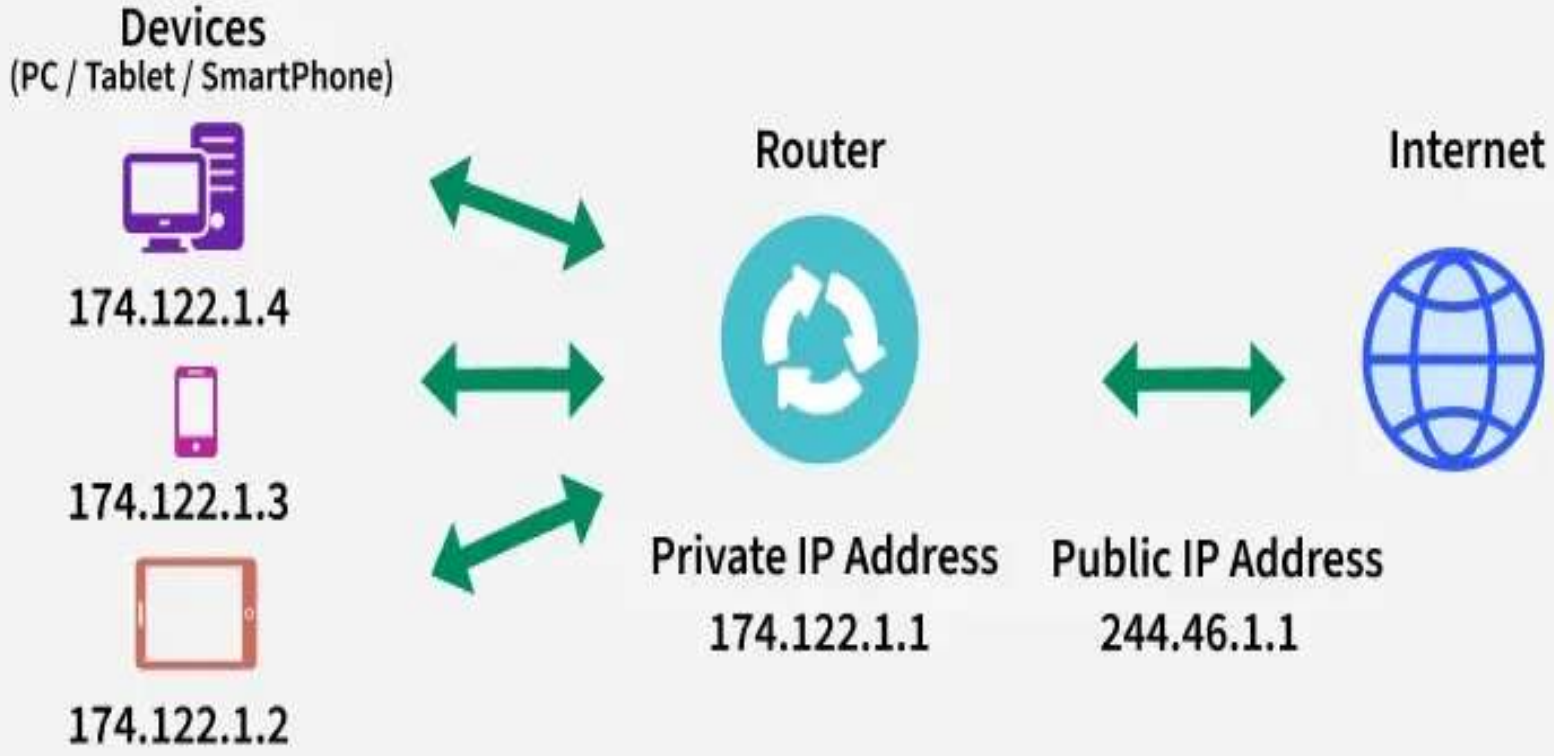


7. Network Address Translation (NAT)

- NAT allows multiple devices in a private network to share a single public IP address for internet access. This is achieved by:
- Translating private IP addresses to a public IP address for outbound traffic.
- Reversing the process for inbound traffic. Benefits of NAT include:
- Conserving IPv4 addresses by reducing the need for unique public IPs for each device.
- Enhancing security by masking internal IP addresses from external networks.

7. Network Address Translation

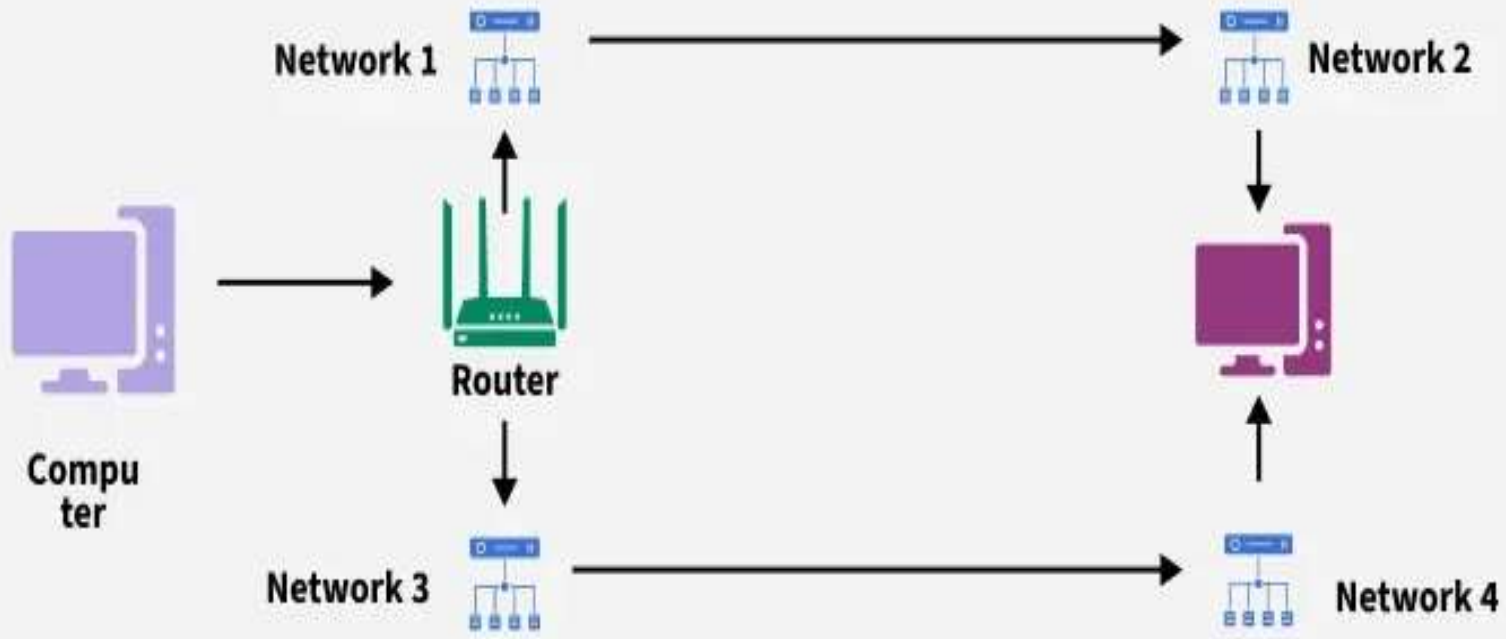
NAT is a method of mapping private IP addresses within a local network to a single public IP address (and vice versa)



8. Routing

- Routing is the process of moving data from one device to another device. These are two other services offered by the network layer. In a network, there are a number of routes available from the source to the destination. **The network layer specifies some strategies which find out the best possible route. This process is referred to as routing.** There are a number of routing protocols that are used in this process and they should be run to help the routers coordinate with each other and help in establishing communication throughout the network.

8. Routing



Implementation of Connectionless Service:

- If connectionless service is offered, packets are injected into the network individually and routed independently of each other.
- The packets are frequently called **datagrams** and the network is called a **datagram network**.
- 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 **Virtual Circuit** and the network is called **Virtual Circuit Network**.
- The algorithm that manages the tables and makes the routing decisions is called the **routing algorithm**

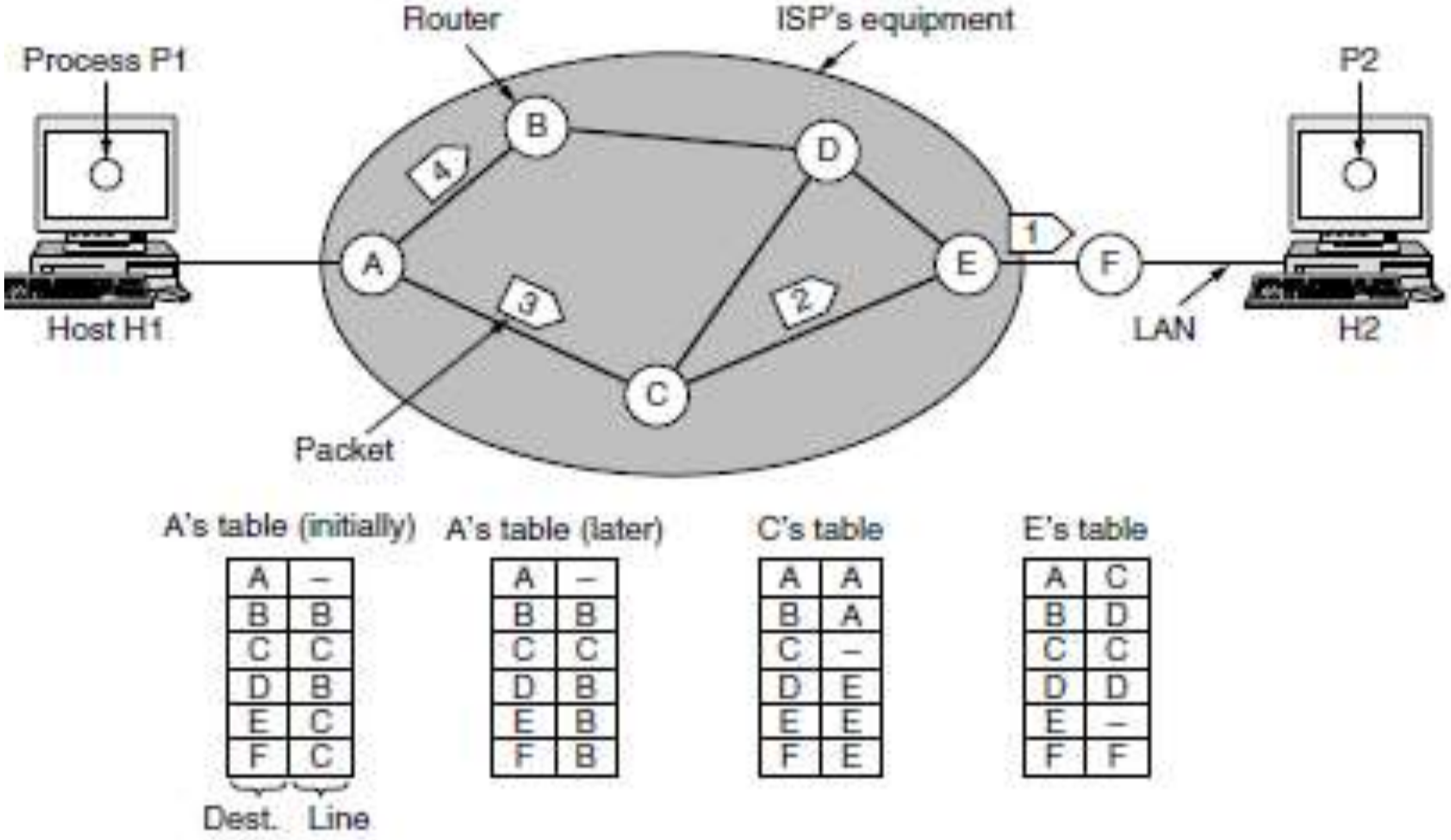


Figure 5-2. Routing within a datagram network.

Implementation of Connection-Oriented Service

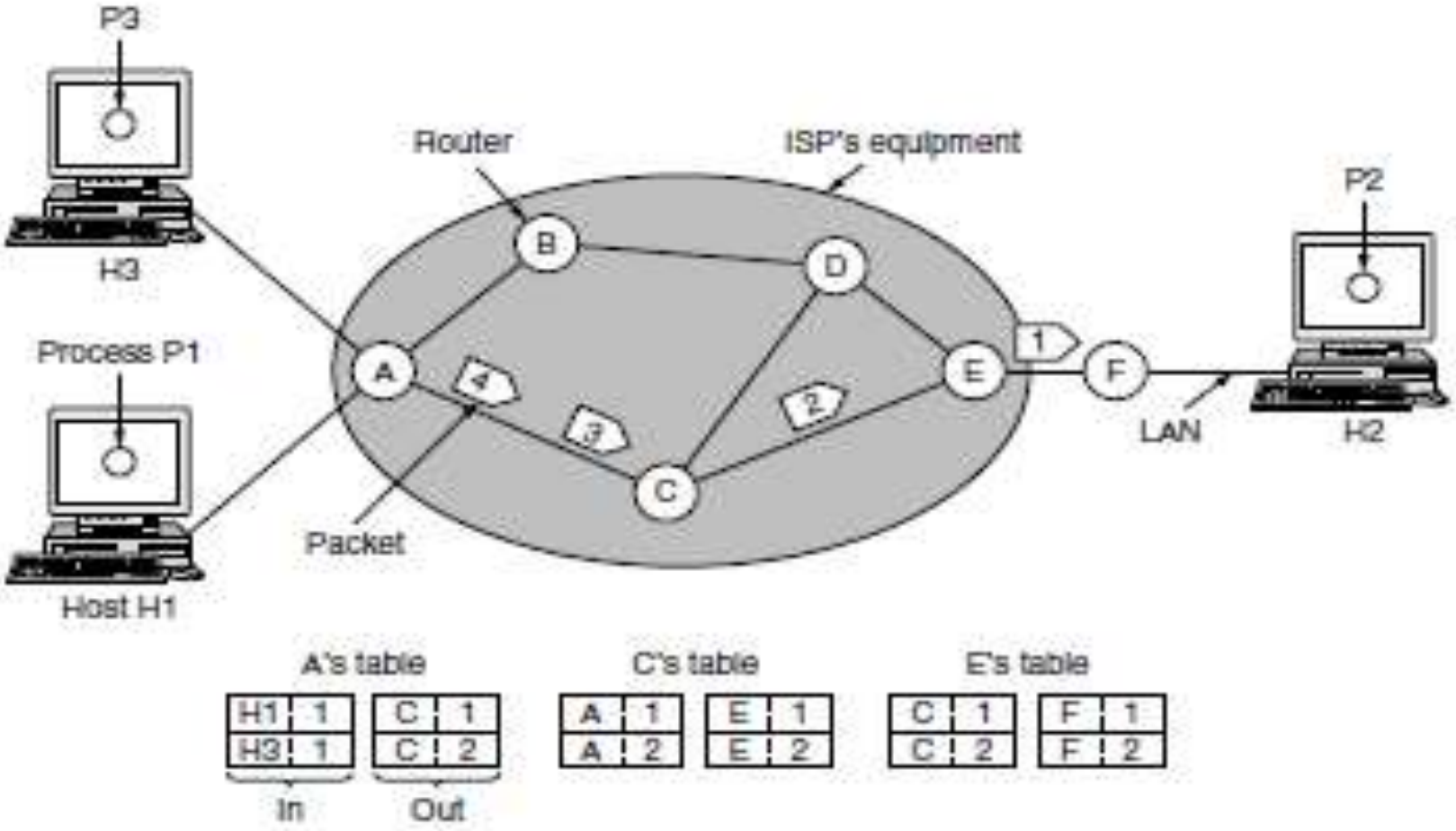


Figure 5-3. Routing within a virtual-circuit network.

Implementation of Connection Oriented service

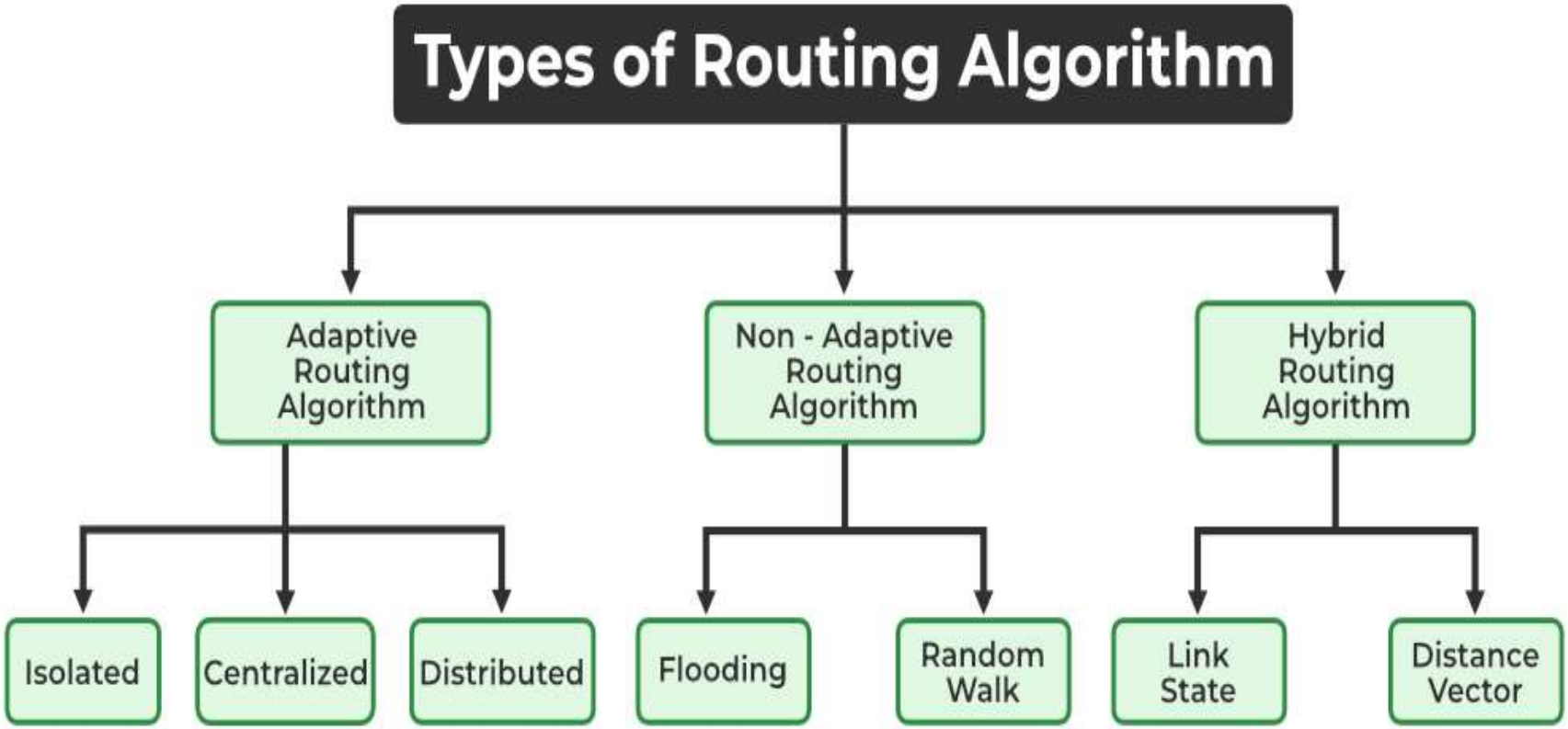
- To use a connection-oriented service, first we establishes a connection, use it and then release it.
- In connection-oriented services, the data packets are delivered to the receiver in the same order in which they have been sent by the sender.
- It can be done in either two ways :
- **Circuit Switched Connection –** A dedicated physical path or a circuit is established between the communicating nodes and then data stream is transferred.
- **Virtual Circuit Switched Connection –** The data stream is transferred over a packet switched network, in such a way that it seems to the user that there is a dedicated path from the sender to the receiver. A virtual path is established here. While, other connections may also be using the same path.

Comparison of Virtual Circuits and Datagram Networks

Issue	Datagram network	Virtual-circuit network
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

Routing Algorithms:

- The algorithm that manages the tables and makes the routing decisions is called the **routing algorithm.**
- The desirable properties of routing algorithms are
  - Correctness,
  - simplicity,
  - robustness,
  - stability,
  - and efficiency are desirable properties in a routing algorithm.


Types of Routing Algorithm
Adaptive Routing Algorithm
Non - Adaptive Routing Algorithm
Hybrid Routing Algorithm
Isolated
Centralized
Distributed
Flooding
Random Walk
Link State
Distance Vector

- Routing algorithms can be grouped into two major classes: **nonadaptive and adaptive**.
- **Nonadaptive algorithms** do not base their routing decisions on any measurements or estimates of the **current topology** and **traffic**.
- Instead, the choice of the route *is computed in advance, offline,* and downloaded to the routers when the network is booted. This procedure is sometimes called **static routing.**
- Examples : Flooding, Shortest Path Routing, and Random Walks.

- **Adaptive algorithms** change their routing decisions to reflect changes in the topology, and sometimes changes in the traffic as well.
- These algorithms get information locally, from adjacent routers or from all routers, when they change the routes.
- That's why there are also known as **Dynamic routing algorithms.**
- Examples : Border Gateway Protocol (BGP), Dynamic Source Routing (DSR), and distance-vector algorithms

BASIS FOR COMPARISON	STATIC ROUTING	DYNAMIC ROUTING
Configuration	It is Manually configure.	It is automatically configure with the help of routing protocols.
Routes	Routes are user defined.	Routes are updated according to change in topology.
Implemented in	It is implemented in Small networks.	It is implemented in Large networks.
Link failure	Link failure obstructs the rerouting.	Link failure doesn't affect the rerouting.
Security	Static routing provides high security.	Dynamic routing is less secure due to sending broadcasts and multicasts.
Additional resources	Additional resources are not required in static routing.	Dynamic routing needs additional resources to store the information.

The Optimality Principle:

- It states that if router J is on the optimal path from router I to router K,
- then the optimal path from J to K also falls along the same route.
- The set of optimal routes to a particular node forms a sink tree.
- 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.

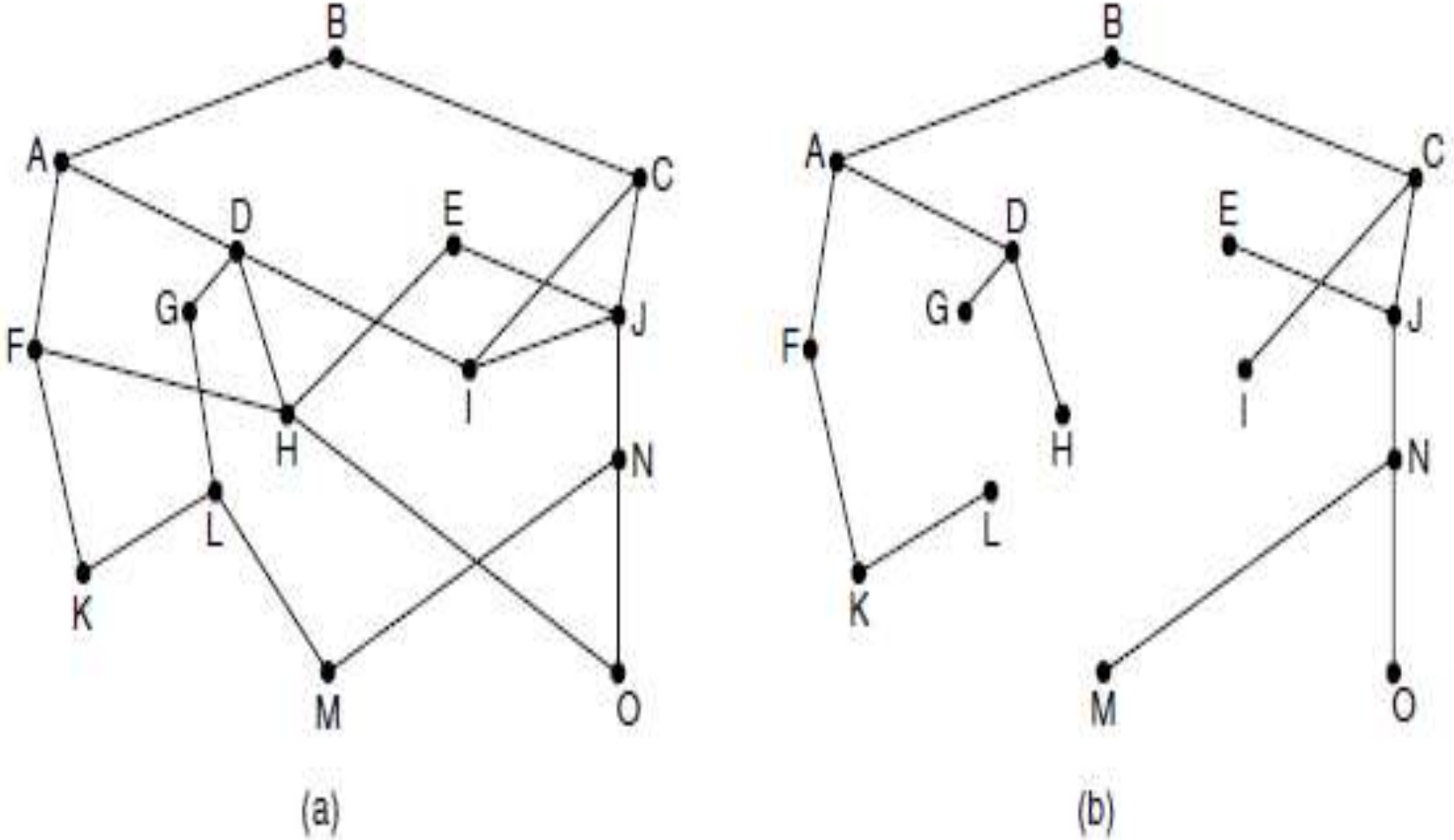


Figure 5-6. (a) A network. (b) A sink tree for router *B*.

Shortest Path Routing Algorithm:

Shortest Path Algorithm:

- The idea is to build a graph of the network, with each node of the graph representing a router and each edge of the graph representing a communication line, or link.
- To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.
- Dijkstra finds the shortest paths between a source and all destinations in the network. Each node is labeled (in parentheses) with its distance from the source node along the best known path.
- Initially, no paths are known, so all nodes are labeled with infinity. As the algorithm proceeds and paths are found, the labels may change, reflecting better paths.

- A label may be either tentative or permanent.
- Initially, all labels are tentative.
- When it is discovered that a label represents the shortest possible path from the source to that node, it is made permanent and never changed thereafter.

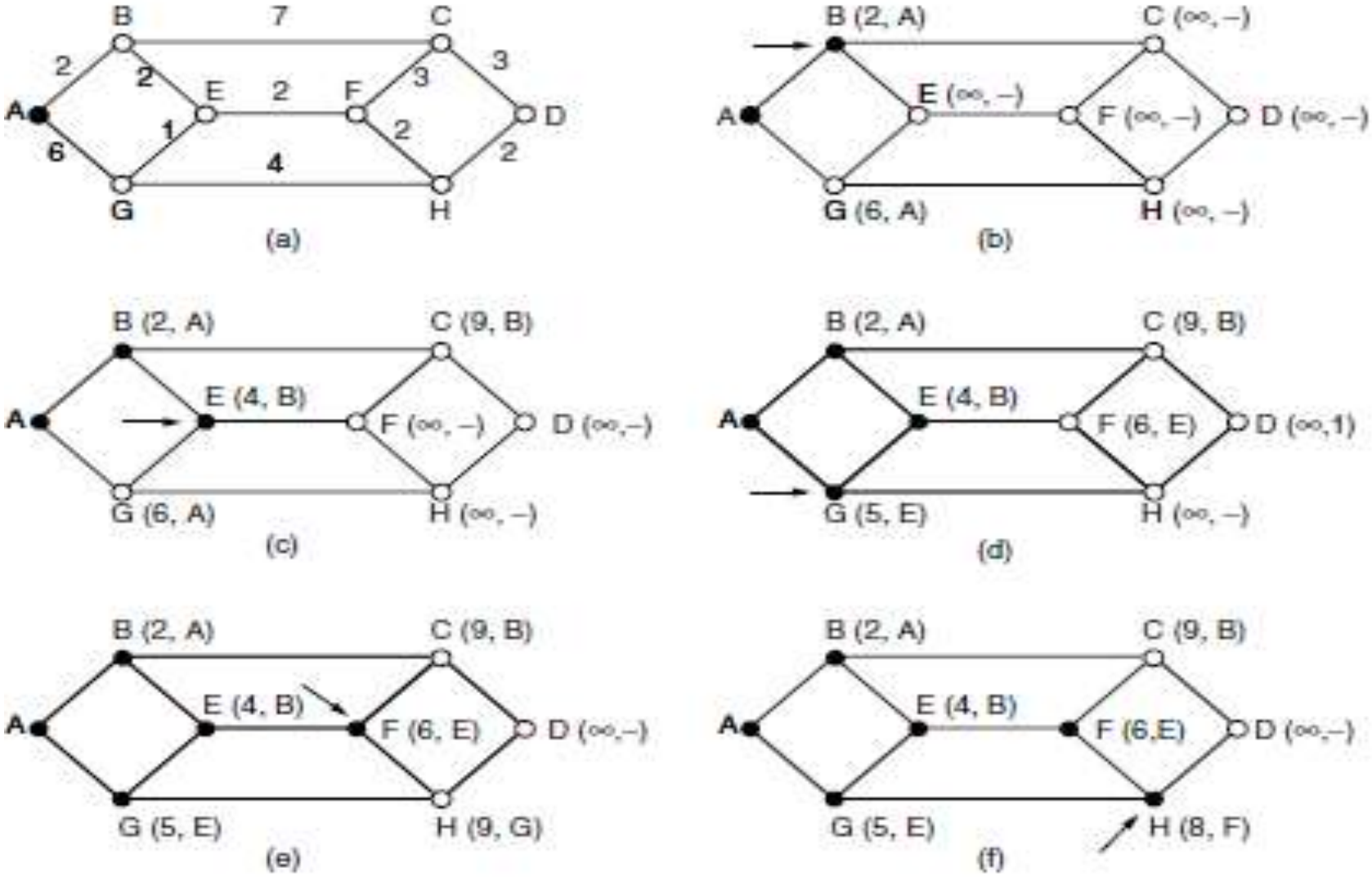


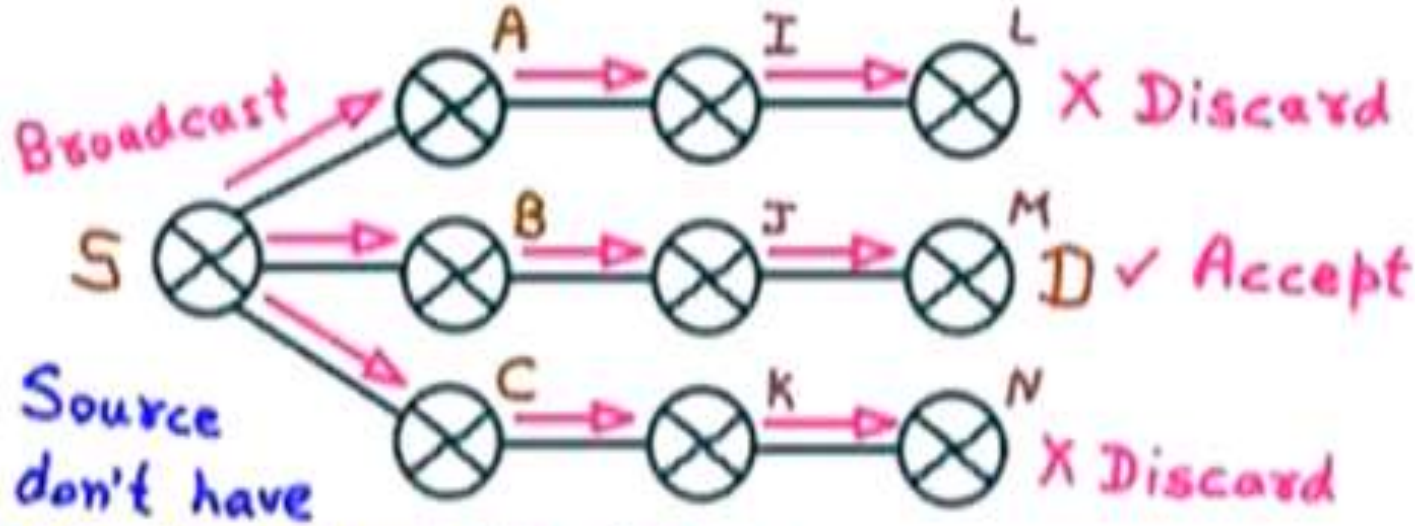
Figure 5-7. The first six steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

Flooding:

- Flooding is a local technique in which every incoming packet is sent out on every outgoing line except the one it arrived on.
- Packet send by node to its every neighbour.
- Flooding generates vast numbers of duplicate packets that reach the destination.
- All nodes will be visited.
- All possible routes will be tried.
- How to eliminate duplicates?

  --Hop Counter.

  --Sequence Number.



Source don't have any idea about Destination

In Flooding every incoming packet is sent out an every outgoing line except the one it arrived on

Disadvantage:
Duplicate Packets

Application:
Military

Distributed Database

Hop Counter:

- Hop counter contained in the header of each packet is decremented at each hop and the packet will be discarded when the counter reaches zero.
- Ideally, the hop counter should be initialized to the length of the path from source to destination.

Sequence Number:

- Source router will put a sequence number in each packet it receives from its hosts.
- Each router then 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

Distance vector Routing

Distance vector Routing:

- Computer networks generally use dynamic routing algorithms that are more complex than flooding, but more efficient because they find shortest paths for the current topology.
- A distance vector routing algorithm operates by having each router maintain a table (i.e., a vector) giving the best known distance to each destination and which linked. These tables are updated by exchanging information with the neighbours.
- Eventually, every router knows the best link to reach each destination.
- Distance vector routing algorithm is also known as **Bellman-Ford routing algorithm.**
- In routing table, each entry has two parts: the preferred outgoing line to use for that destination and an estimate of the distance to that destination.

- The distance might be measured as the number of hops or propagation delay.
- Part (a) shows a network. The first four columns of part (b) show the delay vectors received from the neighbors of router *J.* A claims to have a 12-msec delay to B, a 25-msec delay to C, a 40- msec delay to D, etc.
- Suppose that J has measured or estimated its delay to its neighbors, A, I, H, and K, as 8, 10, 12, and 6 msec, respectively.
- Consider how J computes its new route to router G. It knows that it can get to A in 8 msec, and furthermore A claims to be able to get to G in 18 msec, so J knows it can count on a delay of 26 msec to G if it forwards packets bound for G to A.

- Similarly, it computes the delay to *G via I, H, and K as 41 (31 + 10), 18*
- (6 + 12), and 37 (31 + 6) msec, respectively.

- The best of these values is 18, so it makes an entry in its routing table that the delay to *G is 18 msec and that the route* to use is via *H.*
- The same calculation is performed for all the other destinations, with the new routing table shown in the last column of the figure.
- From J to G JA=8
- JB= A -→JA+AB=8+12=20 JC= A→JA+AC=8+25=33
- H →JH+HB=12+31=43 H→JH+HC=12+19=31
- I →JI+IB=10+36=46 I→JI+IC=10+18=28
- K →JK+KB=6+28=34 K→JK+KC=8+36=42

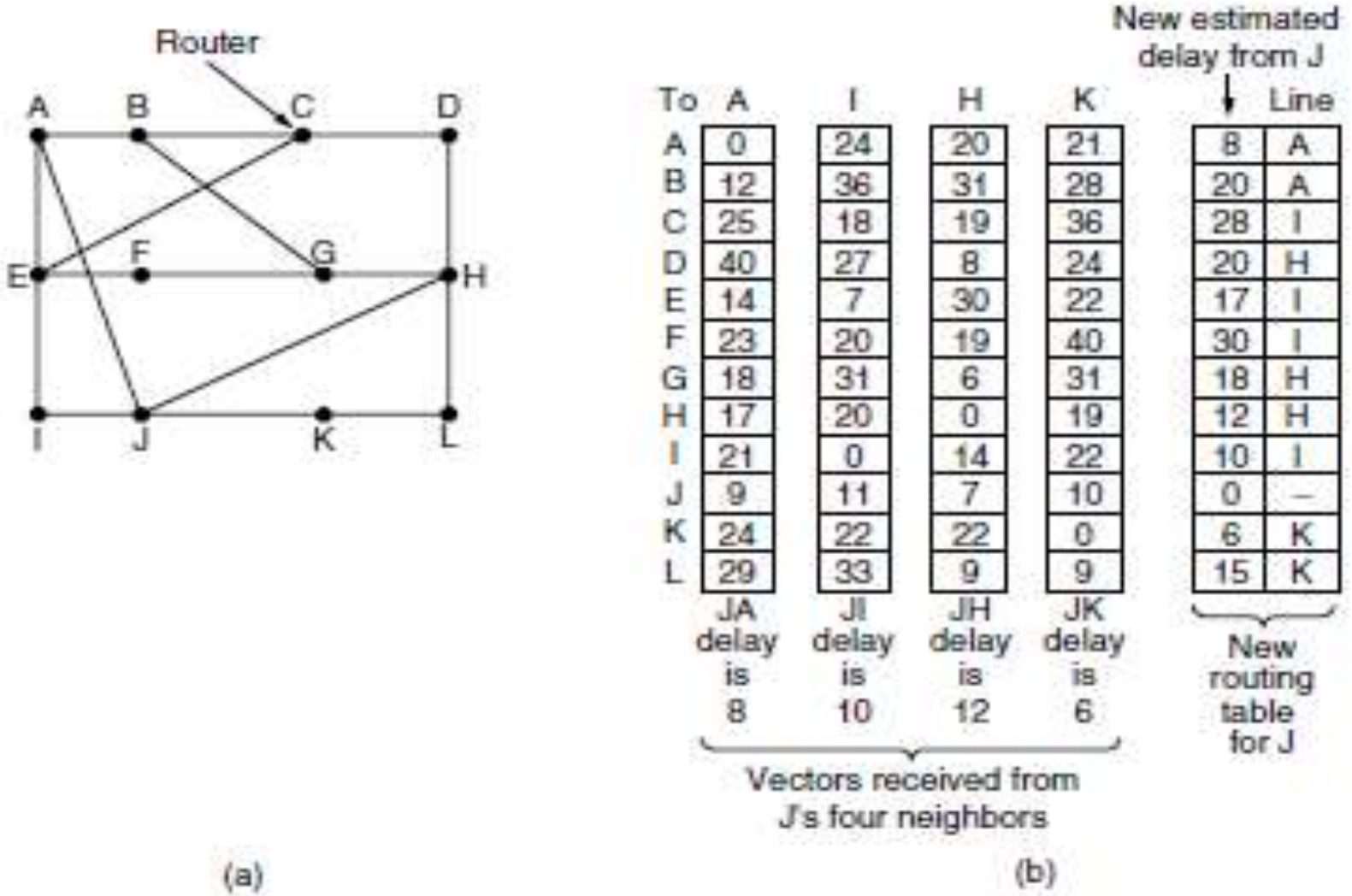


Figure 5-9. (a) A network. (b) Input from *A*, *I*, *H*, *K*, and the new routing table for *J*.

Algorithm Applied

Bellman-Ford Algorithm

$$d_x(y) = \min_v\{c(x, v) + d_v(y)\}$$

Where,

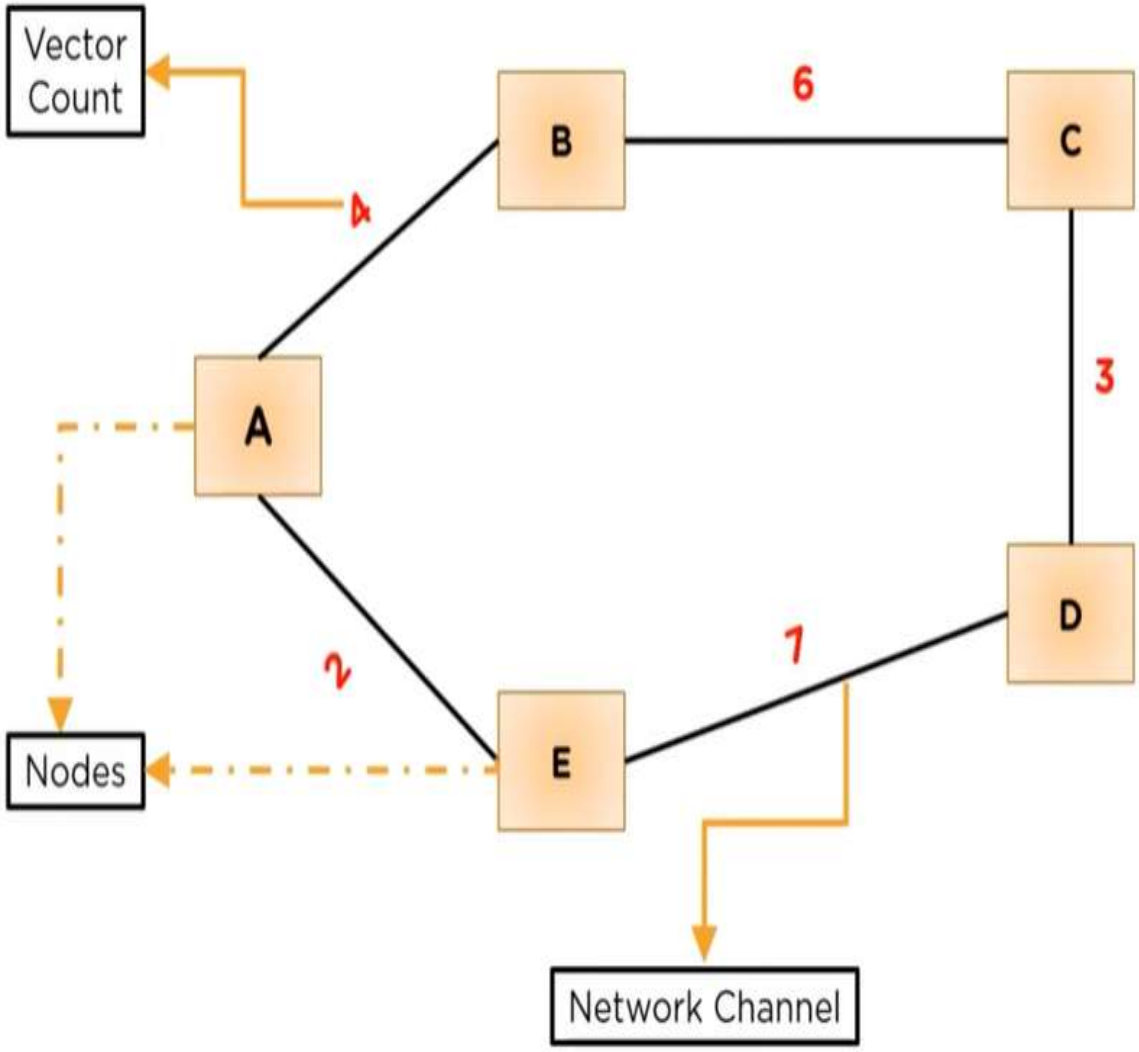
$d_x(y)$ - The least distance from x to y.

$c(x, v)$ - Node x's cost from each of its neighbour v.

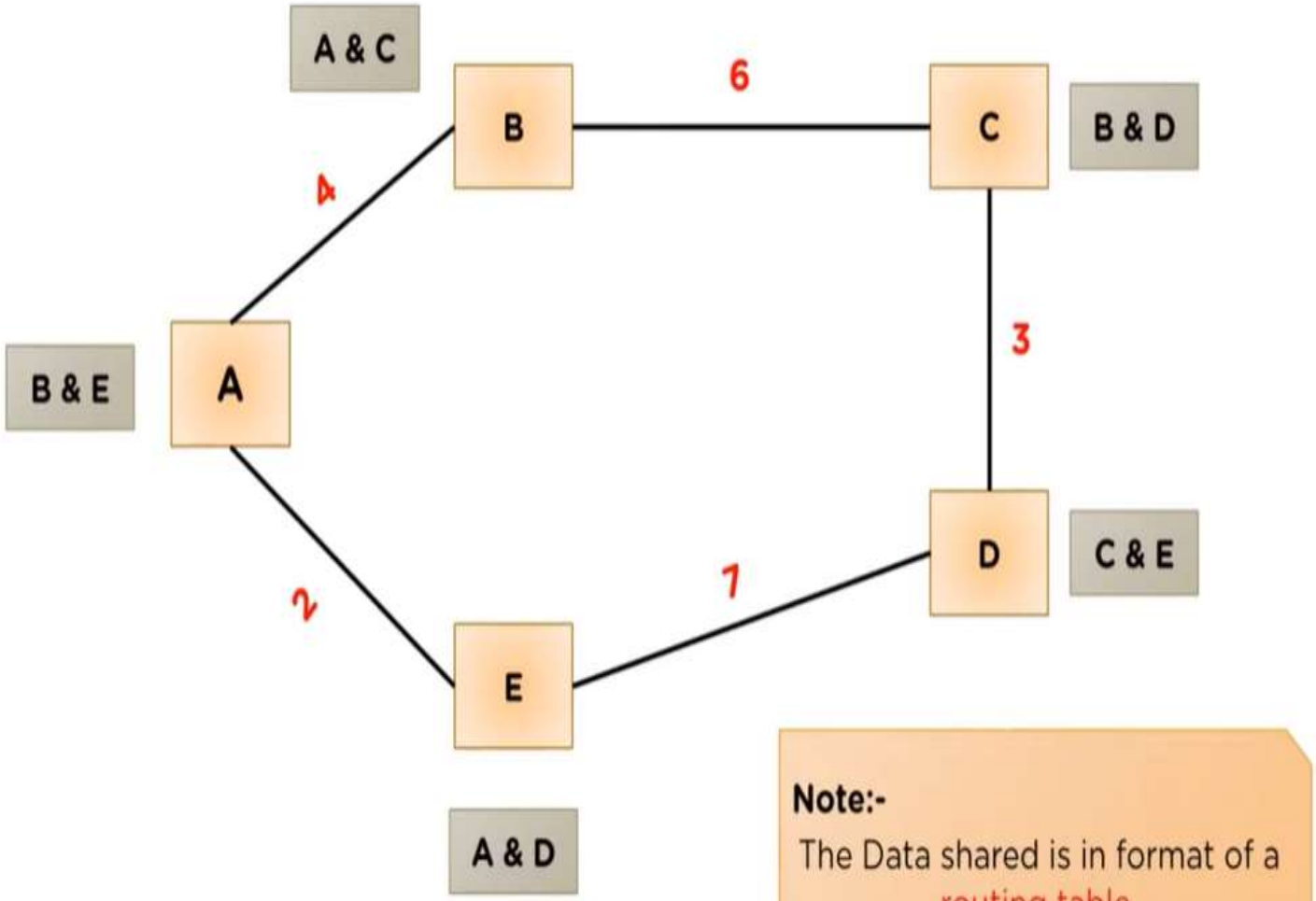
$d_v(y)$ - Distance of each neighbor from initial node.

$\min_v$ - Selecting the minimum distance for the data packet.

Network Example



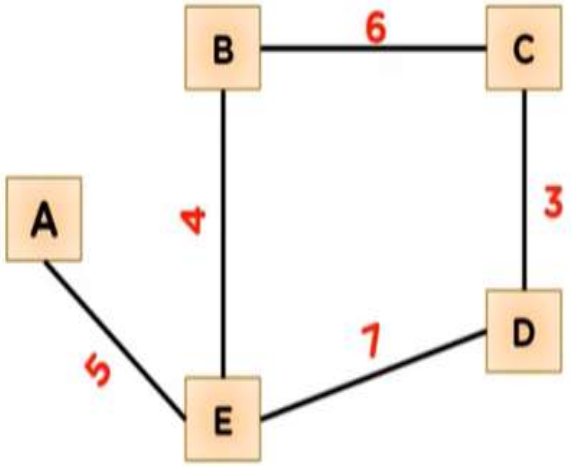
Network Example



Worked Example

Node A

Destination	Vector	Hop
A	0	A
B	∞	-
C	∞	-
D	∞	-
E	5	E



Destination	Vector	Hop
A	∞	-
B	6	B
C	0	C
D	3	D
E	∞	-

Node C

Destination	Vector	Hop
A	5	A
B	4	B
C	∞	-
D	7	D
E	0	E

Node E

Initial Step

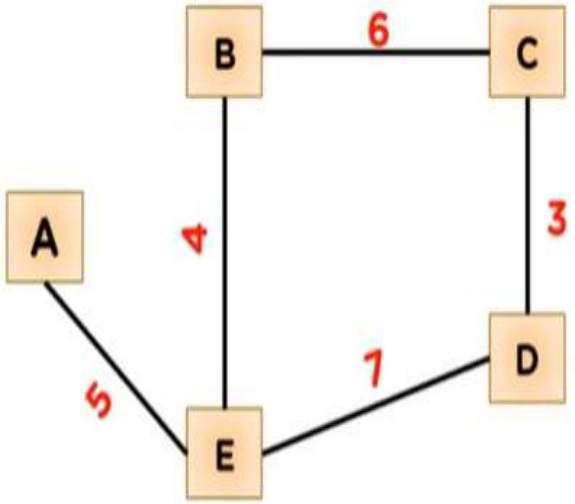
Worked Example

Node A

Destination	Vector	Hop
A	0	A
B	∞	-
C	∞	-
D	∞	-
E	5	E

Node E

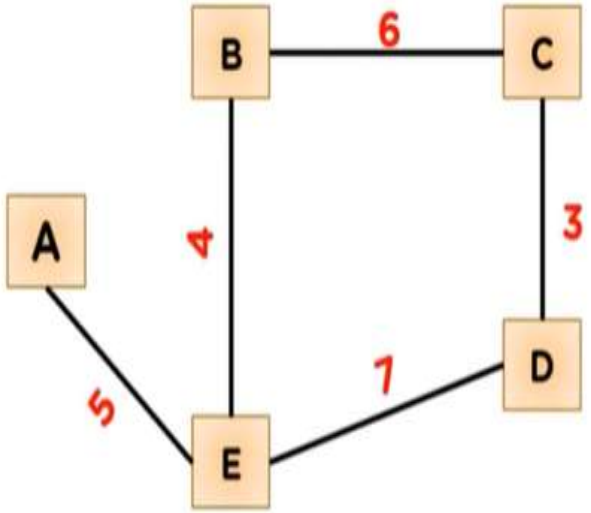
Destination	Vector	Hop
A	5	A
B	4	B
C	∞	-
D	7	D
E	0	E



- A to B:
  (A,E) + (E-B)
  5 + 4
  9
- A to C:
  (A,E) + (E-C)
  5 + ∞
  -
- A to D:
  (A,E) + (E-D)
  5 + 7
  12
- A to E:
  (A,E)
  5

Update Step

Worked Example



Node A

Destination	Vector	Hop
A	0	A
B	9	E
C	∞	-
D	12	E
E	5	E

- A to B:
  (A,E) + (E-B)
  5 + 4
  9

- A to C:
  (A,E) + (E-C)
  5 + ∞
  -

- A to D:
  (A,E) + (E-D)
  5 + 7
  12

- A to E:
  (A,E)
  5

Update Step

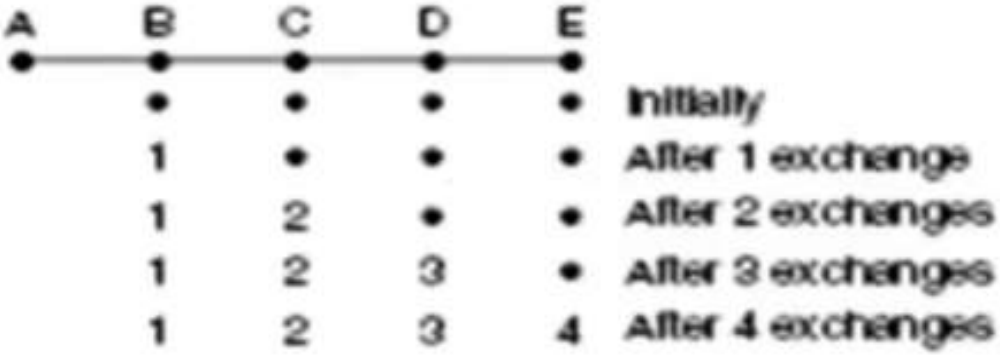
The Count-to-Infinity Problem

- The settling of routes to best paths across the network is called **convergence.**
- Distance vector routing is useful as a simple technique by which routers can collectively compute shortest paths, but it has a serious drawback in practice:
- **Although it converges to the correct answer, it may do so slowly**.
- **In particular, it reacts rapidly to good news, but leisurely to bad news.**

The Count-to-Infinity Problem

(for mentioned example good news propagate in 4 iteration)

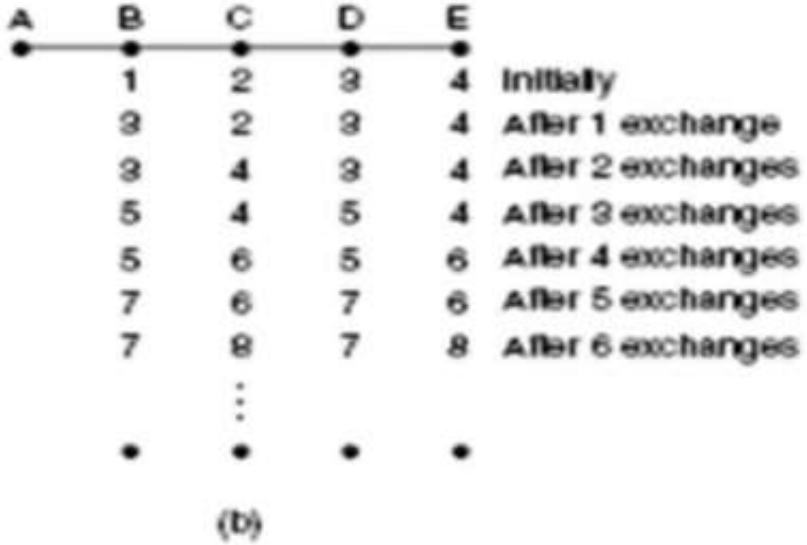
When *A comes up, the other routers learn about it via the vector exchanges.* At the time of the first exchange, *B learns that its left-hand neighbor has zero delay* to *A. B now makes an entry in its routing table indicating that A is one hop away* to the left. All the other routers still think that *A is down.*



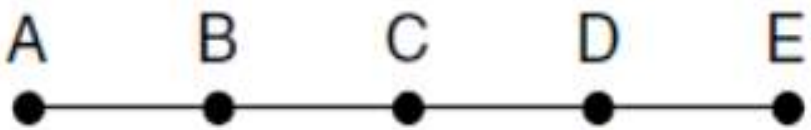
(a)

The Count-to-Infinity Problem

(bad news –link is down, found in so many iteration)



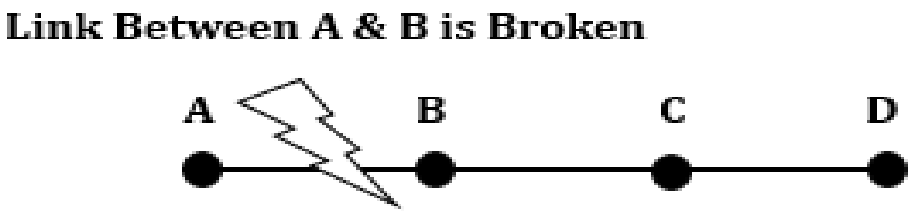
At the first packet exchange, *B does not hear anything from A. Fortunately, C* says "Do not worry; I have a path to *A of length 2." so B will update path A via C. and so on for others also*



1	2	3	4	Initially
3	2	3	4	After 1 exchange
3	4	3	4	After 2 exchanges
5	4	5	4	After 3 exchanges
5	6	5	6	After 4 exchanges
7	6	7	6	After 5 exchanges
7	8	7	8	After 6 exchanges
	⋮			
•	•	•	•	

The Count-to-Infinity Problem:

- Distance vector routing is useful as a simple technique by which routers can collectively compute shortest paths, but it has a serious drawback in practice: although it converges to the correct answer, it may do so slowly.
- In particular, it reacts rapidly to good news, but leisurely to bad news.



	A	B	C	D
A	0,-	1,A	2,B	3,C
B	1,B	0,-	1,B	2,C
C	2,B	1,C	0,-	1,C
D	3,B	2,C	1,D	0,-

- Imagine that the link between A and B is cut.
- At this time, B corrects its table.
- After a specific amount of time, routers exchange their tables, and so B receives C's routing table.
- Since C doesn't know what has happened to the link between A and B, it says that it has a link to A with the weight of 2 (1 for C to B, and 1 for B to A -- it doesn't know B has no link to A).
- B receives this table and thinks there is a separate link between C and A, so it corrects its table and changes infinity to 3 (1 for B to C, and 2 for C to A, as C said).
- Once again, routers exchange their tables.
- When C receives B's routing table, it sees that B has changed the weight of its link to A from 1 to 3, so C updates its table and changes the weight of the link to A to 4 (1 for C to B, and 3 for B to A, as B said).
- This process loops until all nodes find out that the weight of link to A is infinity.

	B	C	D
Sum of Weight to A after link cut	∞, A	2, B	3, C
Sum of Weight to A after 1st updating	3, C	2, B	3, C
Sum of Weight to A after 2nd updating	3, C	4, B	3, C
Sum of Weight to A after 3rd updating	5, C	4, B	5, C
Sum of Weight to A after 4th updating	5, C	6, B	5, C
Sum of Weight to A after 5th updating	7, C	6, B	7, C
Sum of Weight to A after nth updating	.....	....	....
∞	∞	∞	∞

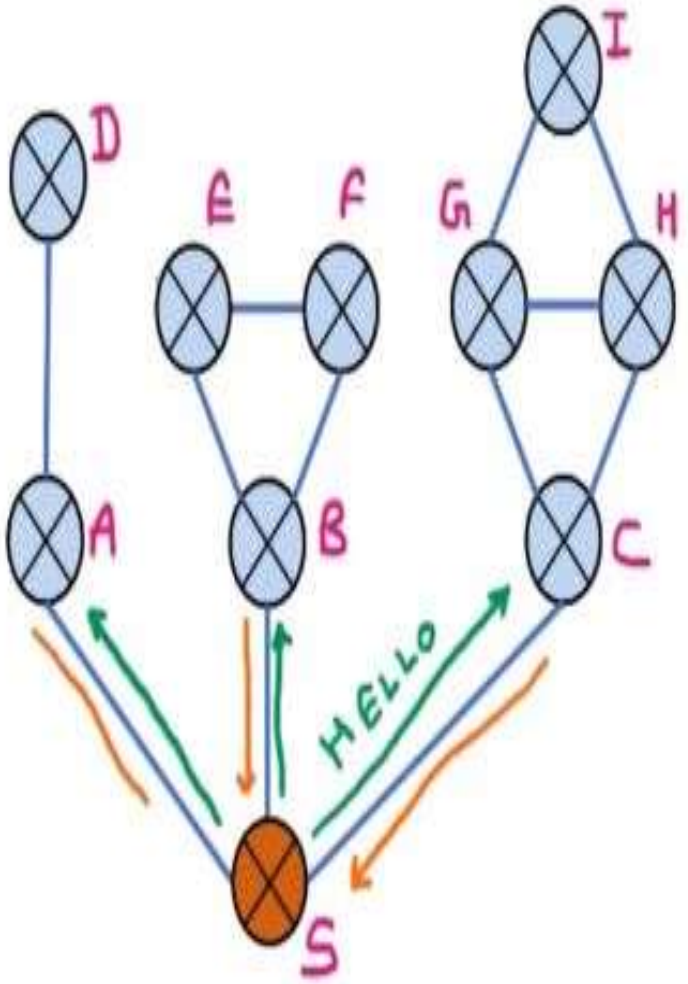
Routing Algorithms

Link State Routing

Each Router do the following:

- Learn their Neighbours & Network Addresses

Send HELLO Packets



Routing Algorithms

Link State Routing

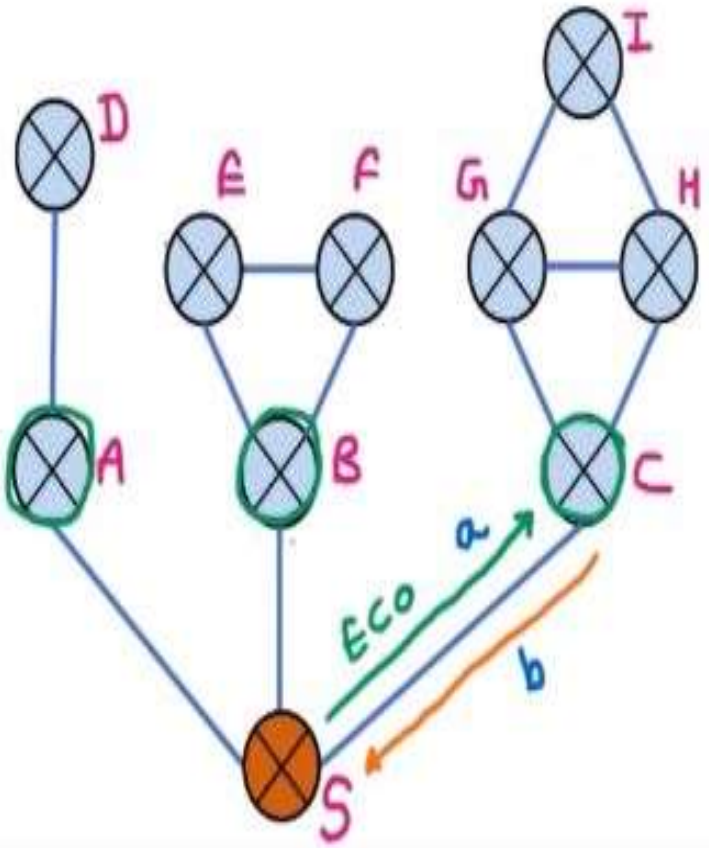
Each Router do the following:

- Learn their Neighbours & Network Addresses

  Send HELLO Packets

- Measure Delay or Cost to each of its Neighbours

  Send ECO Packets



Round Trip Time = $a + b$

Estimated Delay = $\frac{a+b}{2}$

Routing Algorithms

Link State Routing

Each Router do the following:

- Learn their Neighbours & Network Addresses

  Send HELLO Packets

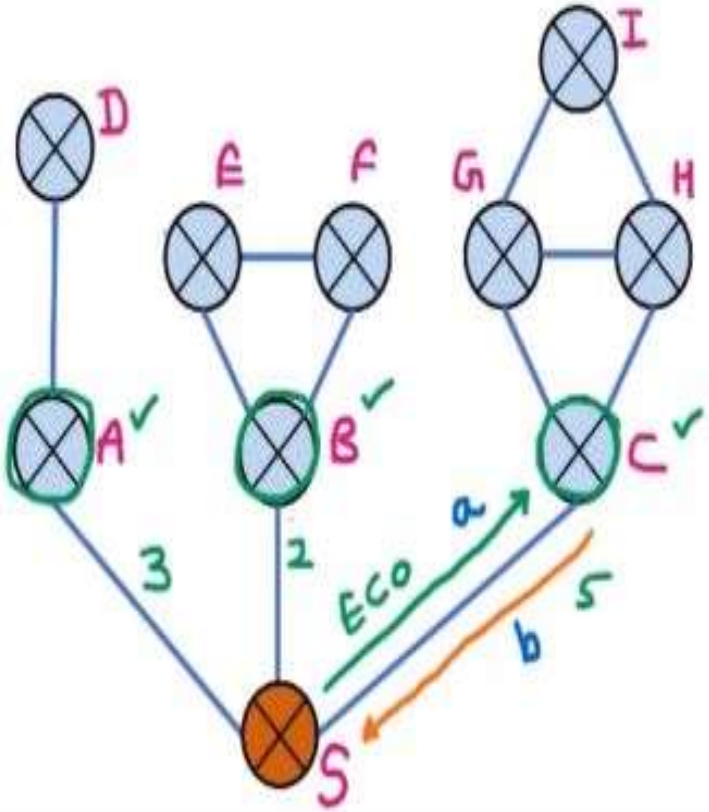
- Measure Delay or Cost to each of its Neighbours

  Send ECO Packets

  Round Trip Time = $a + b$

  Estimated Delay = $\frac{a+b}{2}$

- Construct Packet telling all it has learned



S	
Seq	
Age	
A	3
B	2
C	5

Routing Algorithms

Link State Routing

Each Router do the following:

- Learn their Neighbours & Network Addresses

  Send HELLO Packets

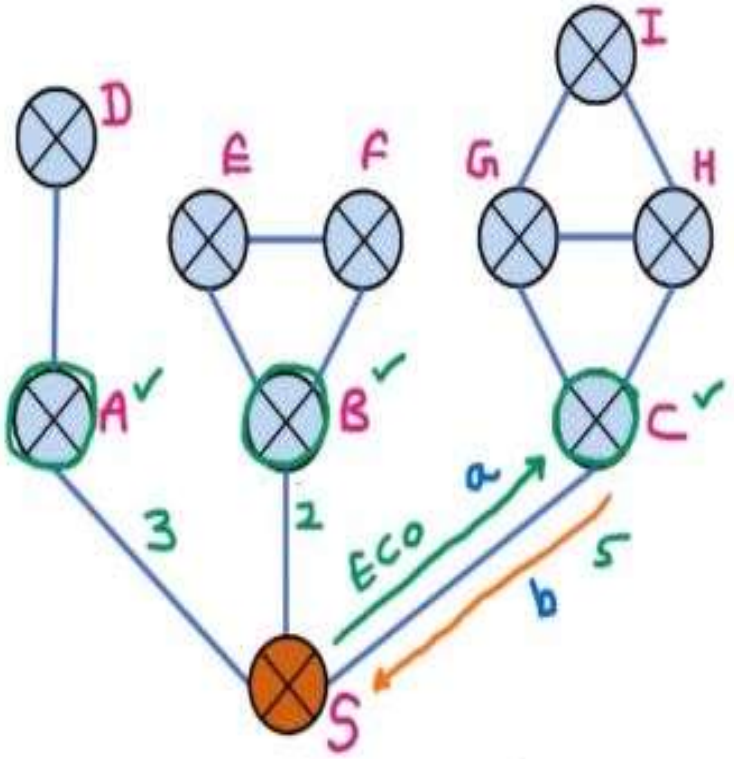
- Measure Delay or Cost to each of its Neighbours

  Send ECO Packets

  Round Trip Time = $a + b$

  Estimated Delay = $\frac{a+b}{2}$

- Construct Packet telling all it has learned



S	
Seq	
Age	
A	3
B	2
C	5

- Send this Packet to all other Routers
- Compute Shortest Path to every other Router

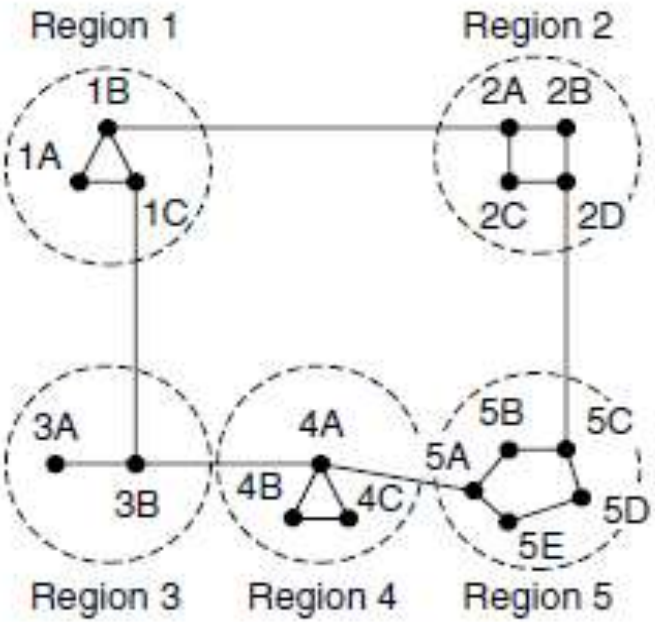
Hierarchical Routing

Hierarchical Routing:

- At a certain point, the network may grow to the point where it is no longer feasible for every router to have an entry for every other router, so the routing will have to be done hierarchically, as **it is in the telephone network.**
- When hierarchical routing is used, the routers are divided into **regions**. Each router knows all the details about how to route packets to destinations within its own region but knows nothing about the internal structure of other regions.
- When different networks are interconnected, it is natural to regard each one as a separate region to free the routers in one network from having to know the topological structure of the other ones.

- Figure 5-14 gives a quantitative example of routing in a two-level hierarchy with five regions.
- The full routing table for router 1A has 17 entries, as shown in Fig. 5-14(b).
- When routing is done hierarchically, as in Fig. 5-14(c), there are entries for all the local routers, as before, but all other regions are condensed into a single router, so all traffic for region 2 goes via the 1B-2A line, but the rest of the remote traffic goes via the 1C-3B line.
- Hierarchical routing has reduced the table from 17 to 7 entries.
- There is a penalty to be paid: increased path length. For example, the best route from 1A to 5C is via region 2, but with hierarchical routing all traffic to region 5 goes via region 3, because that is better for most destinations in region 5.

- For example, consider a network with 720 routers. If there is no hierarchy, each router needs 720 routing table entries.
- If the network 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.
- If a three-level hierarchy is chosen, 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 cluster, and 7 entries for distant clusters, for a total of 25 entries.
- **Kamoun and Kleinrock (1979) discovered that the optimal number of levels for an N router network is ln N.**



(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.

Broadcast Routing:

Broadcast Routing:

- In some applications, hosts need to send messages to many or all other hosts.
- For example, a service distributing weather reports, stock market updates, or live radio programs might work best by sending to all machines and letting those that are interested read the data.
- Sending a packet to all destinations simultaneously is called **broadcasting.**
- The first broadcasting method is simply sending distinct packet to each destination.
- This method wastes the bandwidth and it is slow , but it also requires the source to have a complete list of all destinations.
- The second broadcast method is **multidestination routing**, in which each packet contains a list of destinations.

- The router generates a new copy of the packet for each output line.
- The next broadcast routing technique is flooding.
- Flooding generates too many packets and consumes too much bandwidth.
- The fourth broadcast algorithm uses sink trees for initiating broadcast.
- The idea for broadcasting is **reverse path forwarding**.
- This being the case, the router forwards copies of it onto all links except the one it arrived on.
- If, however, the broadcast packet arrived on a link other than the preferred one for reaching the source, the packet is discarded as a likely duplicate.

- Sink trees are spanning trees.
- A spanning tree is a subset of the network that includes all the routers but contains no loops.
- If each router knows which of its lines belong to the spanning tree, it can copy an incoming broadcast packet onto all the spanning tree lines except the one it arrived on.
- This method makes excellent use of bandwidth, generating the absolute minimum number of packets necessary to do the job.
- The principal advantage of reverse path forwarding is that it is efficient while being easy to implement.

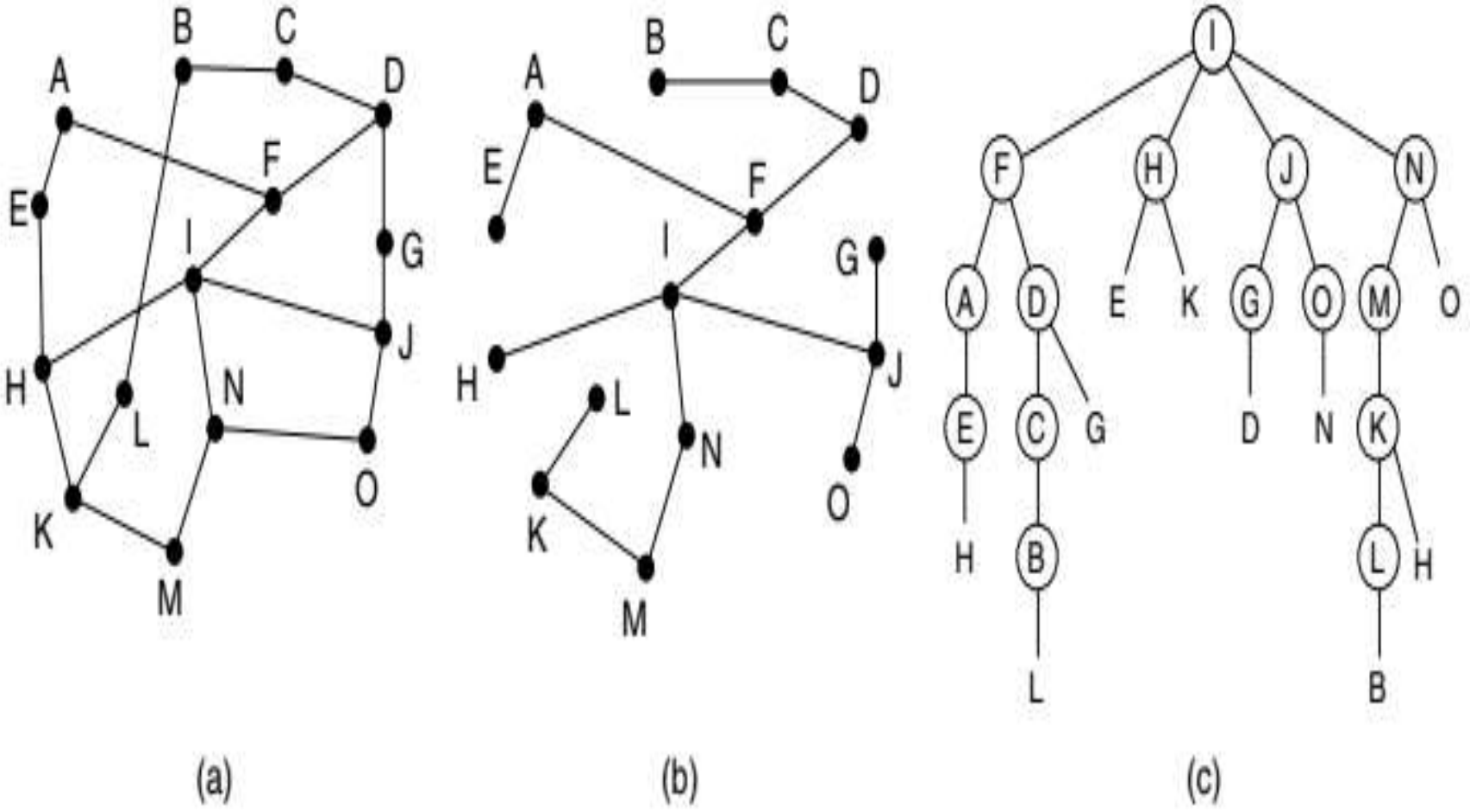


Figure 5-15. Reverse path forwarding. (a) A network. (b) A sink tree. (c) The tree built by reverse path forwarding.

- An example of reverse path forwarding is shown in Fig. 5-15. Part (a) shows a network, part (b) shows a sink tree for router I of that network, and part (c) shows how the reverse path algorithm works.
- On the first hop, I sends packets to F, H, J, and N, as indicated by the second row of the tree. Each of these packets arrives on the preferred path to I (assuming that the preferred path falls along the sink tree) and is so indicated by a circle around the letter.
- On the second hop, eight packets are generated, two by each of the routers that received a packet on the first hop and five of these arrive along the preferred line.
- After five hops and 24 packets, the broadcasting terminates, compared with four hops and 14 packets had the sink tree been followed exactly.

Multicast Routing:

- Broadcasting a packet is wasteful, if the receivers are not supposed to see it.
- Sending a message to a group is called multicasting, and the routing algorithm used is called multicast routing.
- Multicasting requires group management, need to create and destroy groups.
- To do multicast routing each router computes a spanning tree covering all other routers.
- In the below figure we have two groups group1 and group2.
- Some routers are attached to one or more groups.
- Figure (b) is spanning tree for the left most router

- When a process sends a multicast packet to a group, the first router examines its spanning tree and prunes it, removing all lines that do not lead to hosts that are members of the group.
- In our example, Figure (c) shows the pruned spanning tree for group 1.
- Figure(d) shows the pruned spanning tree for group 2. Multicast packets are forwarded only along the appropriate spanning tree.

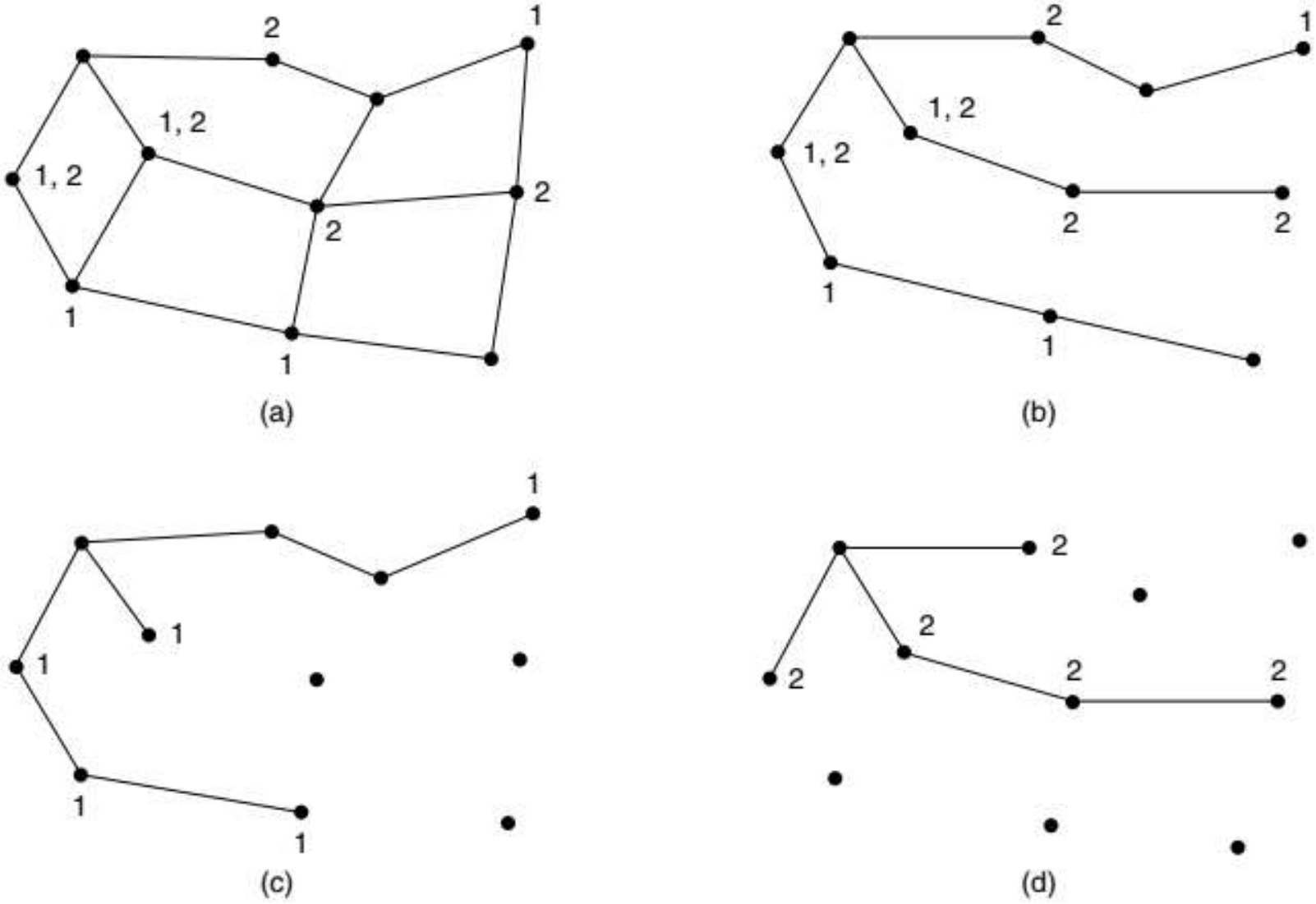
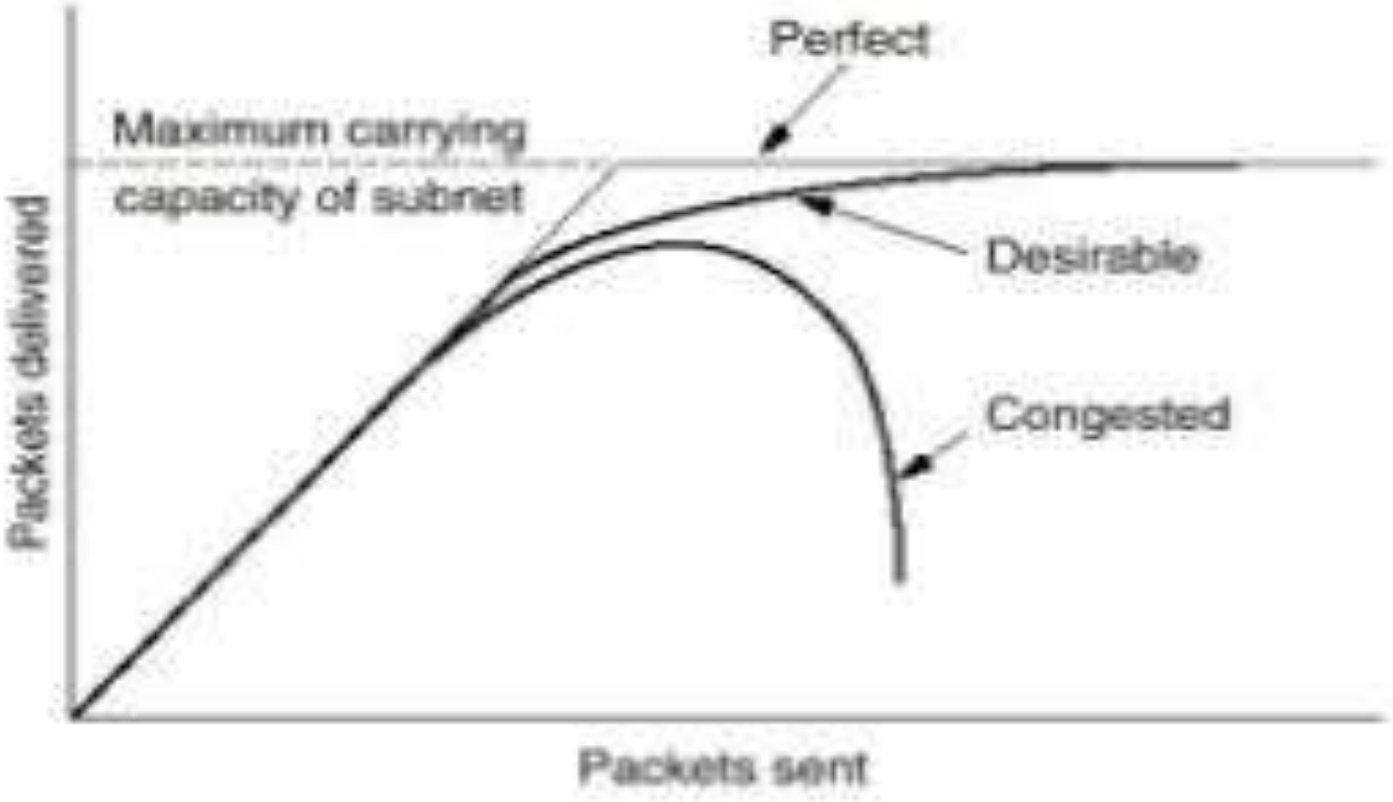


Figure 5-16. (a) A network. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1. (d) A multicast tree for group 2.

Congestion Control Algorithms:

Congestion Control :

- When too many packets are present in the subnet, performance degrades. This situation is called congestion.
- Congestion can be brought on by several factors. If all of a sudden, streams of packets begin arriving on three or four input lines and all need the same output line, a queue will build up. If there is insufficient memory to hold all of them, packets will be lost.
- Slow processors can also cause congestion.
- Similarly, low-bandwidth lines can also cause congestion.


Perfect
Maximum carrying capacity of subnet
Desirable
Congested
Packets delivered
Packets sent

Causes of Congestion

- Congestion occurs when a router receives data faster than it can send it
  - Insufficient bandwidth
  - Slow hosts
  - Data simultaneously arriving from multiple lines destined for the same outgoing line.
- The system is not balanced
  - Correcting the problem at one router will probably just move the bottleneck to another router.

Congestion Control versus Flow Control

- Flow control
  - controls point-to-point traffic between sender and receiver
  - e.g., a fast host sending to a slow host
- Congestion Control
  - controls the traffic throughout the network

General Principles of Congestion Control:

Complex problems of Computer Networks can be viewed from a control point of view. This approach leads to dividing all solutions into two groups: open loop and closed loop.

Open Loop Congestion Control

- Open loop congestion control include deciding when to accept new traffic, deciding when to discard packets and which ones, and making scheduling decisions at various points in the network. The congestion control is handled either by the source or the destination.
- In contrast, **closed loop** solutions are based on the concept of a feedback loop. 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.

1. Monitor the system to detect when and where congestion occurs

- The percentage of all packets discarded due to lack of buffer space, the average queue lengths.
- The number of packets that time out and are retransmitted, the average packet delay.

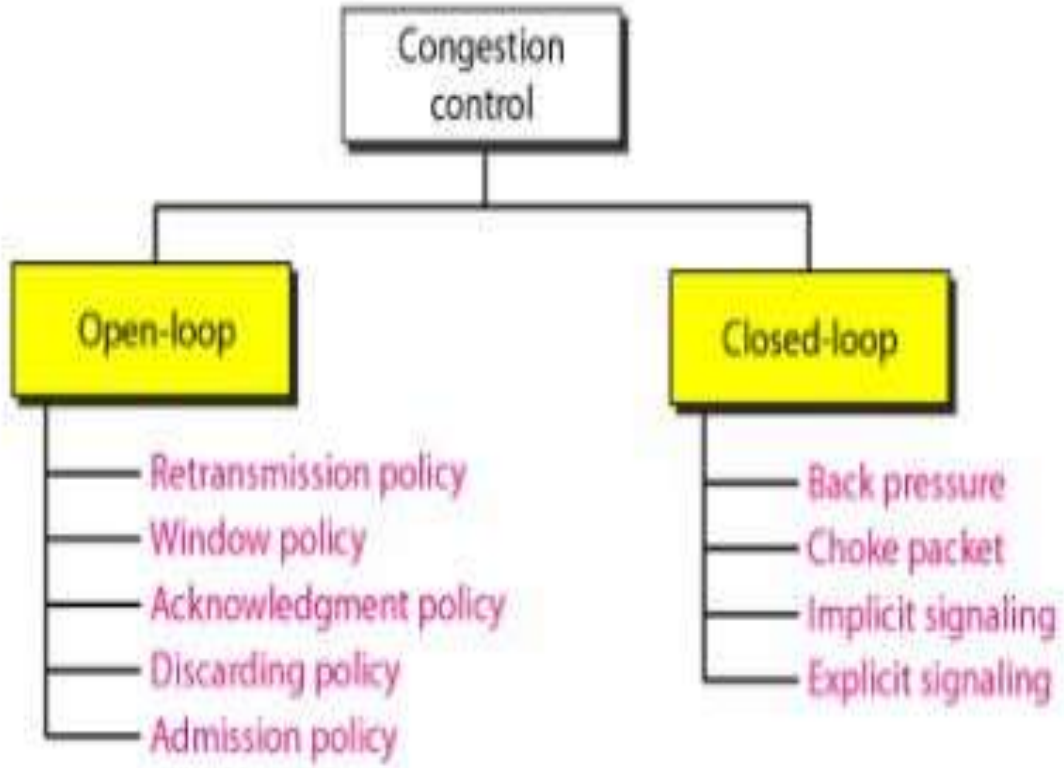
2, **Pass this information to places where action can be taken**

- Send a packet to the traffic source or sources, announcing the problem.
- A bit or field can be reserved in every packet for routers to fill in whenever congestion gets above some threshold level. When a router detects congested state, it fills in the field in all outgoing packets, to warn the neighbors.
- Sending probe packets out to explicitly ask about congestion. This information can then be used to route traffic around problem areas.

3. Adjust system operation to correct the problem:

- The presence of congestion means that the load is (temporarily) greater than the resources (in part of the system) can handle.
- Two solutions come to mind: increase the resources or decrease the load.
- Increasing the resources is not always possible, the only way then to beat back the congestion is to decrease the load.

Congestion Prevention Policies:



Congestion Prevention Policies:

Policies adopted by open loop congestion control :

Retransmission Policy :

It is the policy in which retransmission of the packets are taken care. To prevent congestion, retransmission timers must be designed to prevent congestion and also able to optimize efficiency.

Window Policy :

The type of window at the sender side may also affect the congestion.

Selective repeat window should be adopted as it sends the specific packet that may have been lost because **GoBackN** increases duplication of packets.

Discarding Policy :

A good discarding policy adopted by the routers will help the routers from preventing congestion.

Acknowledgment Policy :

Since acknowledgement are also the part of the load in network, the acknowledgment policy imposed by the receiver may also affect congestion.

Admission Policy :

In admission policy a mechanism should be used to prevent congestion.

Congestion Control in Virtual Circuit subnets

- 1. Admission Control: If congestion has been signaled , no more virtual circuits are set up until the problem has gone away
- 2. Allow new virtual circuits but carefully route all new virtual circuits around problem area. For example the following diagram shows the alternate virtual circuit set up when congestion in the path
- 3. Negotiate an agreement between the host and subnet when virtual circuit is set up. The agreement specifies the volume and shape of the traffic, quality of service required, and other parameters.

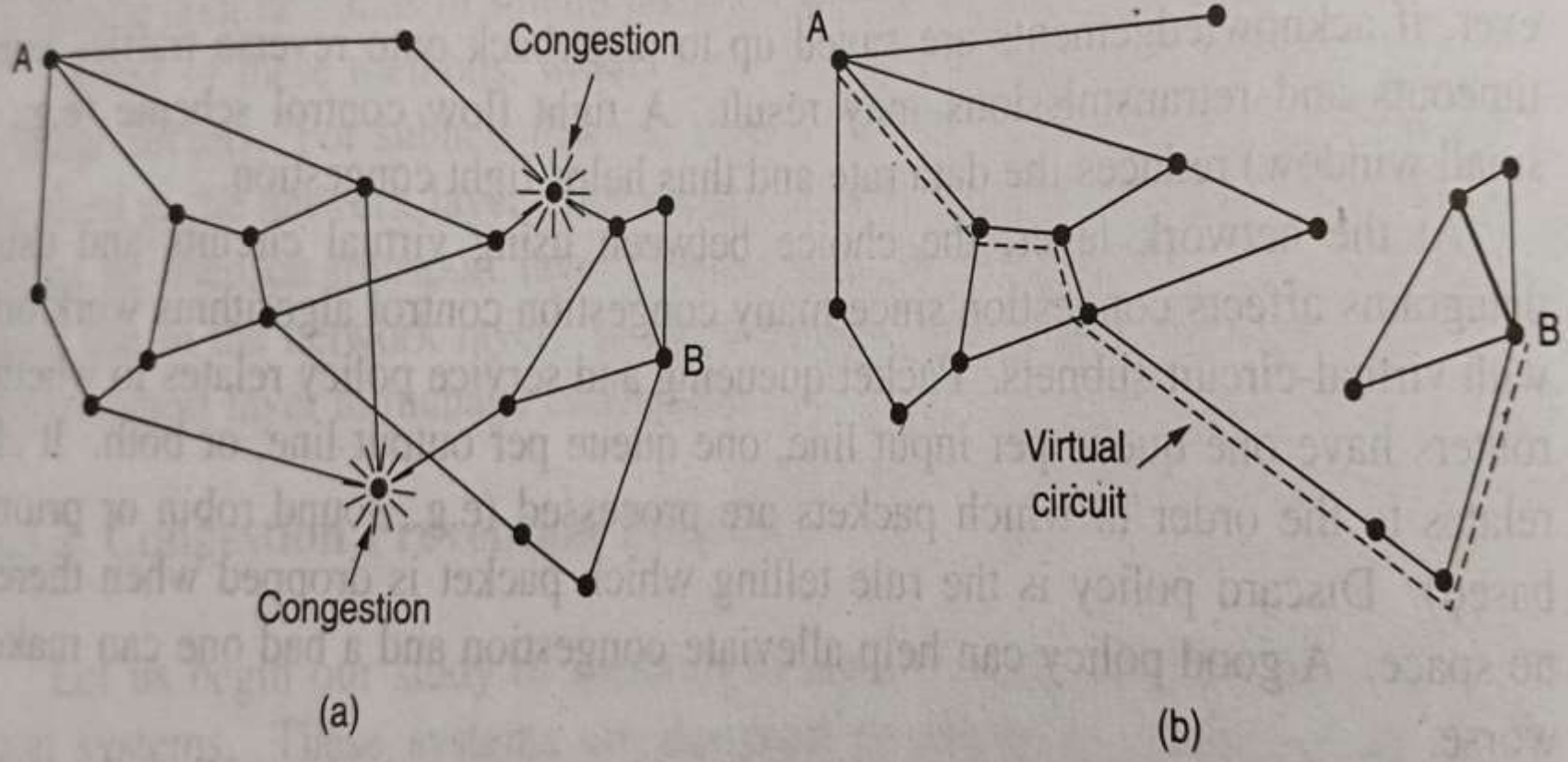


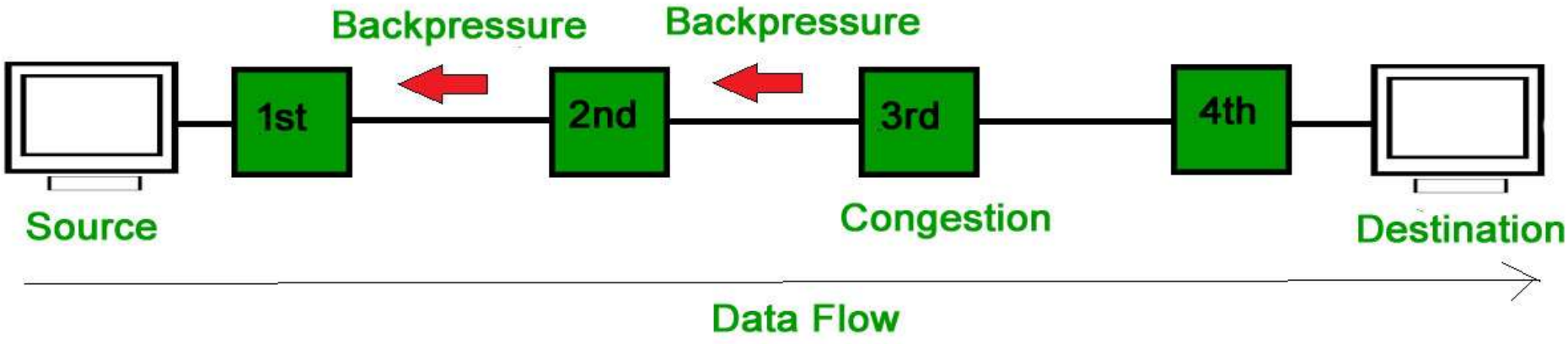
Figure 5-27. (a) A congested subnet. (b) A redrawn subnet that eliminates the congestion. A virtual circuit from *A* to *B* is also shown.

Closed loop congestion control:(Feed Back)

It is used to treat or alleviate congestion after it happens. Several techniques are used by different protocols; some of them are:

Backpressure :

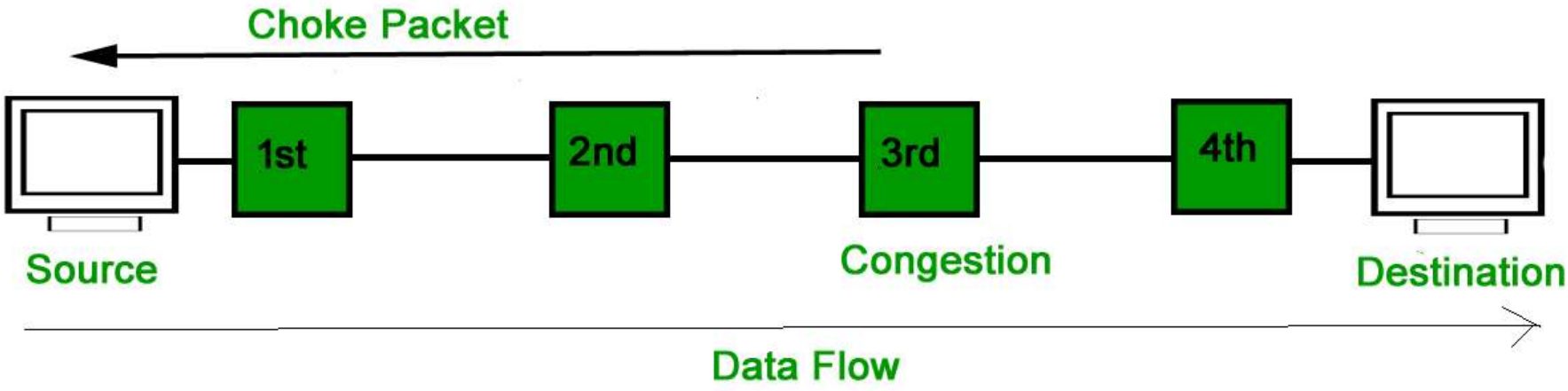
Backpressure is a technique in which a congested node stops receiving packets from upstream node. This may cause the upstream node or nodes to become congested and reject receiving data from above nodes. Backpressure is a node-to-node congestion control technique that propagate in the opposite direction of data flow.



Warning Bit/ Backpressure

- A special bit in the packet header is set by the router to warn the source when congestion is detected.
- The bit is copied and piggy-backed on the ACK and sent to the sender.
- The sender monitors the number of ACK packets it receives with the warning bit set and adjusts its transmission rate accordingly.

- **Choke Packet Technique :** Choke packet technique is applicable to both virtual networks as well as datagram subnets. A choke packet is a packet sent by a node to the source to inform it of congestion. Each router monitors its resources and the utilization at each of its output lines. Whenever the resource utilization exceeds the threshold value which is set by the administrator, the router directly sends a choke packet to the source giving it a feedback to reduce the traffic. The intermediate nodes through which the packets has traveled are not warned about congestion.



Hop-by-Hop Choke Packet

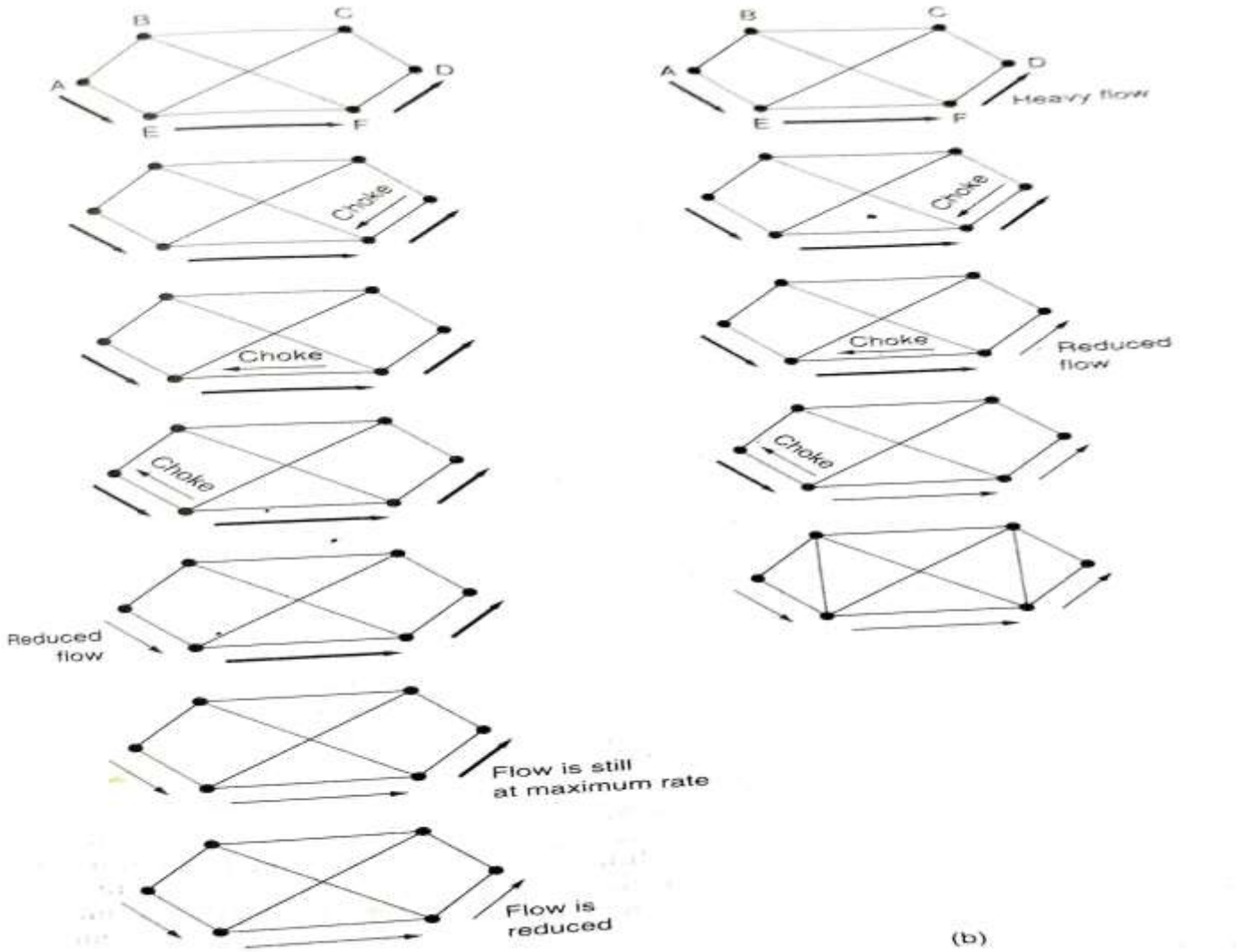


Figure 5-28. (a) A choke packet that affects only the source. (b) A choke packet that affects each hop it passes through.

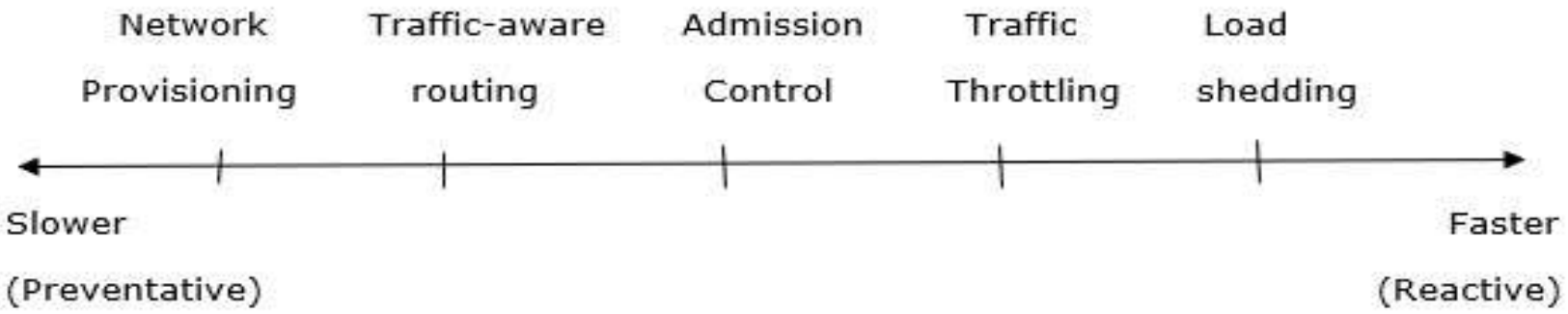
Implicit signaling: There is no communication between the congested nodes and the source. The source guesses that there is congestion in a network.

Explicit signaling: If a node experiences congestion it can explicitly sends a packet to the source or destination to inform about congestion. The difference between choke packet and explicit signaling is that the signal is included in the packets that carry data rather than creating different packet.

- Explicit signaling can occur in either forward or backward direction.
- **Forward Signaling :** In forward signaling, a signal is sent in the direction of the congestion. The destination is warned about congestion. The receiver in this case adopt policies to prevent further congestion.
- **Backward Signaling :** In backward signaling, a signal is sent in the opposite direction of the congestion. The source is warned about congestion and it needs to slow down.

Approaches to Congestion Control

- There are some approaches for congestion control over a network which are usually applied on different time scales to either prevent congestion or react to it once it has occurred



Time scale of approaches to congestion control

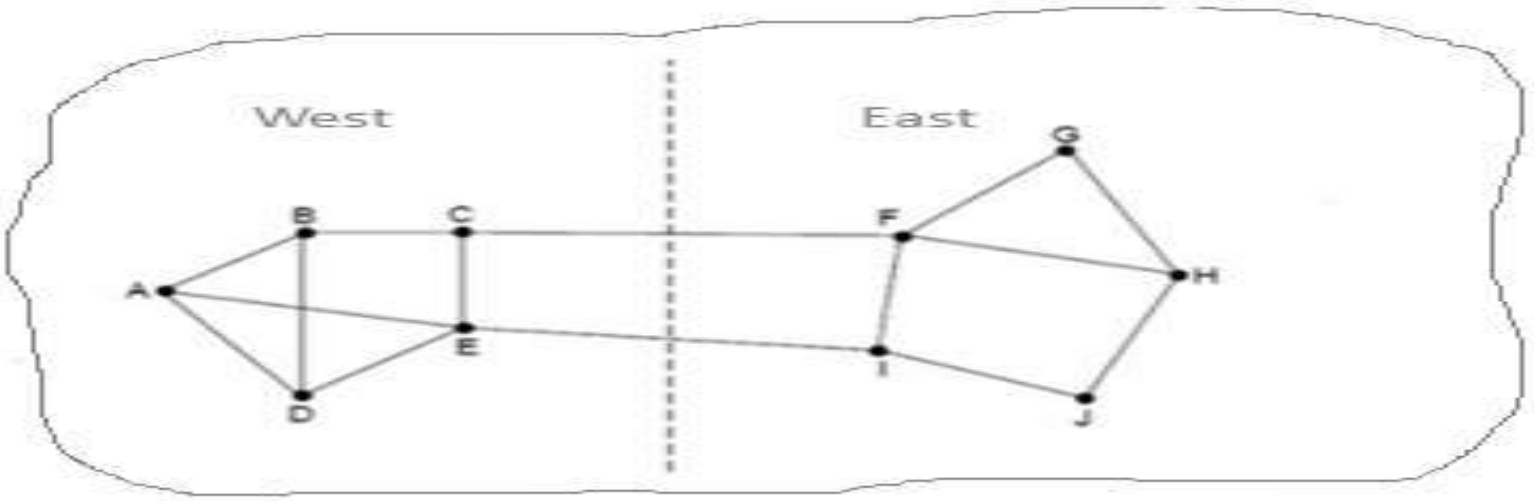
- **Step 1** – The basic way to avoid congestion is to build a network that is well matched to the traffic that it carries. If more traffic is directed but a low-bandwidth link is available, definitely congestion occurs.

- **Step 2** – Sometimes resources can be added dynamically like routers and links when there is serious congestion. This is called provisioning, and which happens on a timescale of months, driven by long-term trends.
- **Step 3** – To utilise most existing network capacity, routers can be tailored to traffic patterns making them active during daytime when network users are using more and sleep in different time zones.
- **Step 4** – Some of local radio stations have helicopters flying around their cities to report on road congestion to make it possible for their mobile listeners to route their packets (cars) around hotspots. This is called traffic aware routing.
- **Step 5** – Sometimes it is not possible to increase capacity. The only way to reduce the congestion is to decrease the load. In a virtual circuit network, new connections can be refused if they would cause the network to become congested. This is called admission control.

- **Step 6** – Routers can monitor the average load, queueing delay, or packet loss. In all these cases, the rising number indicates growing congestion. The network is forced to discard packets that it cannot deliver. The general name for this is Load shedding. The better technique for choosing which packets to discard can help to prevent congestion collapse.

Traffic Aware Routing :

- Traffic awareness is one of the approaches for congestion control over the network. The basic way to avoid congestion is to build a network that is well matched to the traffic that it carries. If more traffic is directed but a low-bandwidth link is available, congestion occurs.
- The main goal of traffic aware routing is to identify the best routes by considering the load, set the link weight to be a function of fixed link bandwidth and propagation delay and the variable measured load or average queuing delay.
- Least-weight paths will then favour paths that are more lightly loaded, remaining all are equal.



Explanation

Step 1 − Consider a network which is divided into two parts, East and West both are connected by links CF and EI.

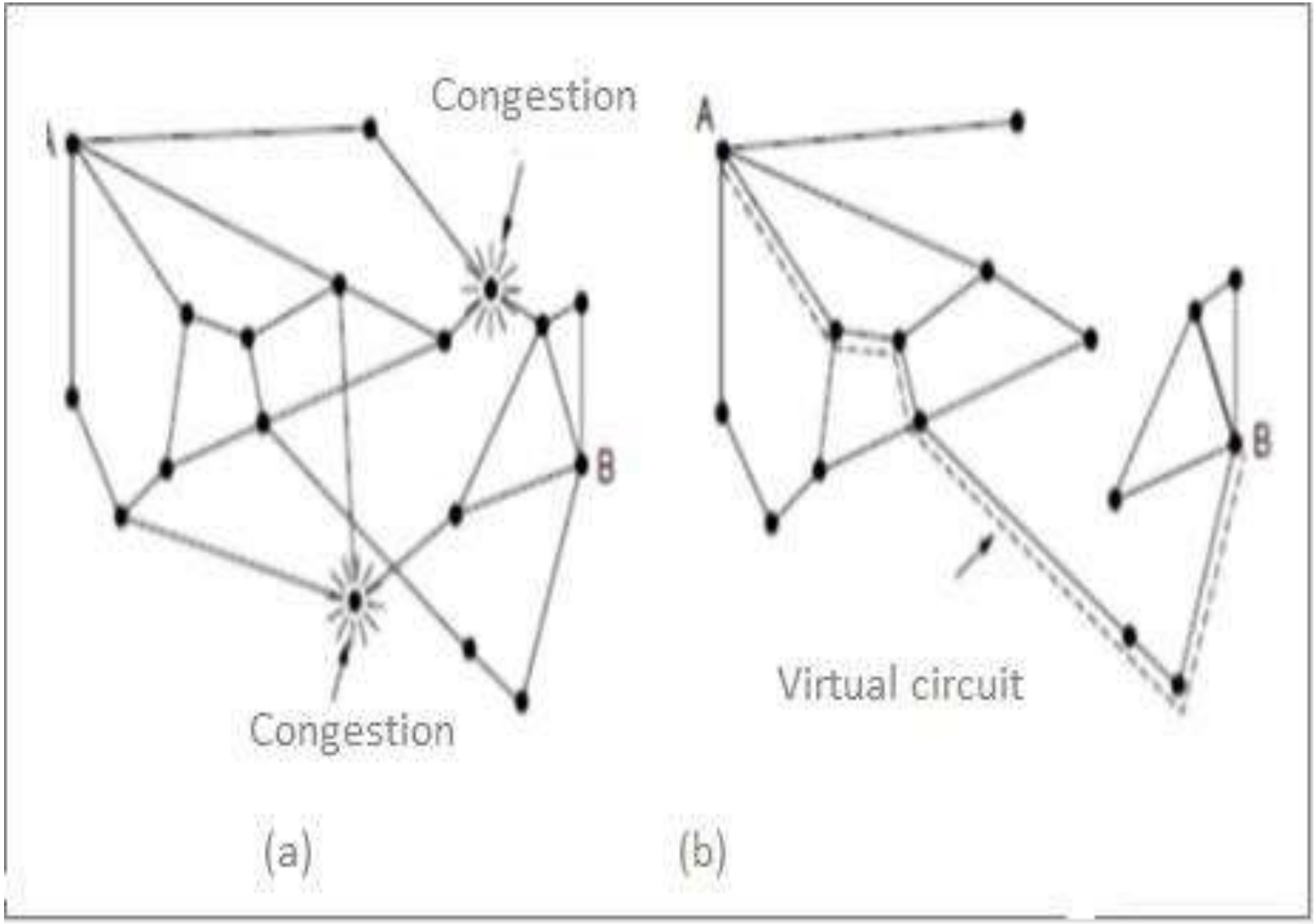
Step 2 − Suppose most of the traffic in between East and West is using link CF, and as a result CF link is heavily loaded with long delays. Including queueing delay in the weight which is used for shortest path calculation will make EI more attractive.

Step 3 − After installing the new routing tables, most of East-West traffic will now go over the EI link. As a result in the next update CF link will appear to be the shortest path.

- **Step 4** – As a result the routing tables may oscillate widely, leading to erratic routing and many potential problems.
- **Step 5** – If we consider only bandwidth and propagation delay by ignoring the load, this problem does not occur. Attempts to include load but change the weights within routing scheme to shift traffic across routes arrow range only to slow down routing oscillations.
- **Step 6** – Two techniques can contribute for successful solution, which are as follows –
- Multipath routing
- The routing scheme to shift traffic across routes.

Admission Control :

- Admission control in computer networks is a congestion prevention technique that restricts new connections by checking for sufficient available resources before granting access to the network. It ensures network demand does not exceed supply, maintaining performance by guaranteeing a certain Quality of Service (QoS) for admitted flows, preventing congestion, packet delay, and loss. Admission control mechanisms use flow specifications and resource reservation to manage bandwidth and buffer space, accepting or rejecting new requests based on whether the requested QoS can be met without violating existing commitments.

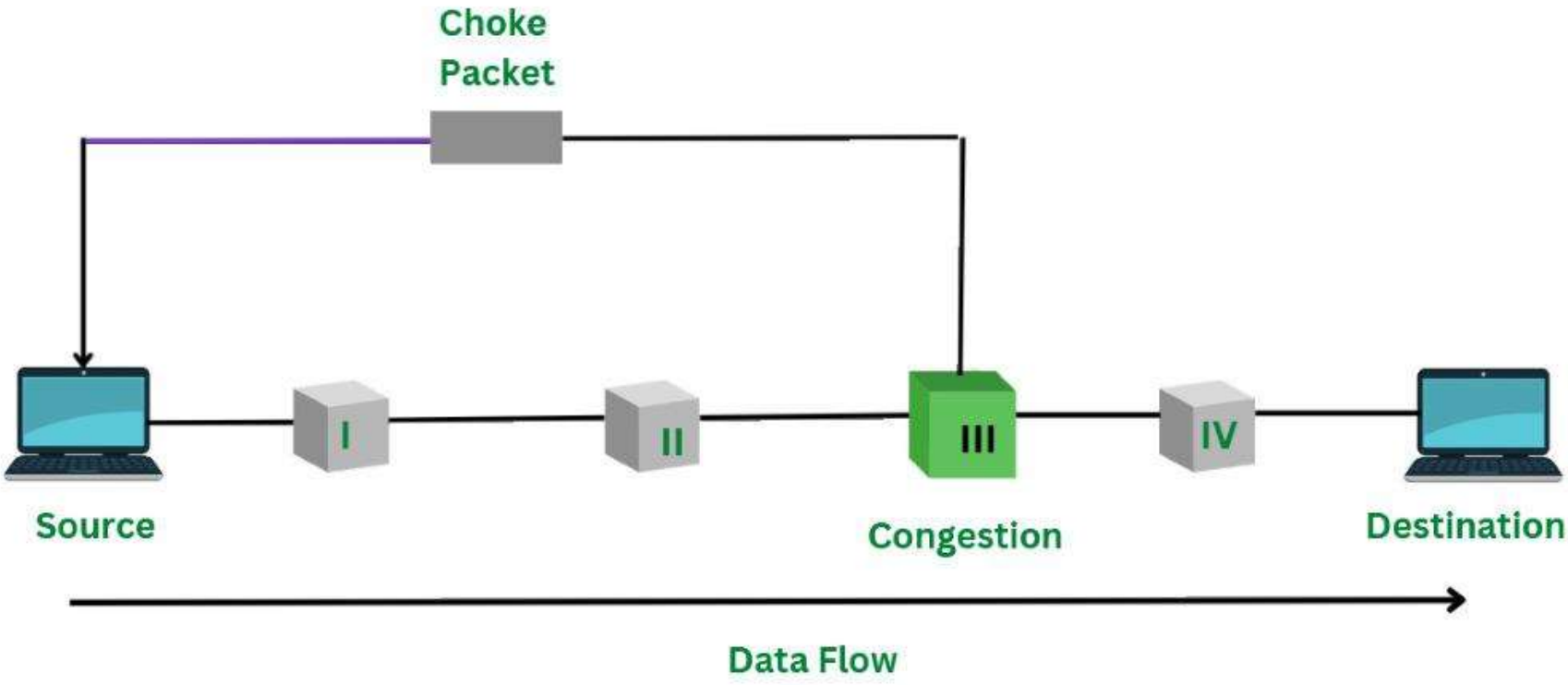

Congestion
B
Congestion
(a)
A
B
Virtual circuit
(b)

Traffic Throttling:

- Traffic Throttling is an approach used to avoid congestion. In networks and the internet, the senders try to send as much traffic as possible as the network can readily deliver. In a network when congestion is approaching it should tell the senders of packets to slow down them. Traffic Throttling can be used in virtual circuit networks and datagram networks. Various approaches are used for throttling traffic.

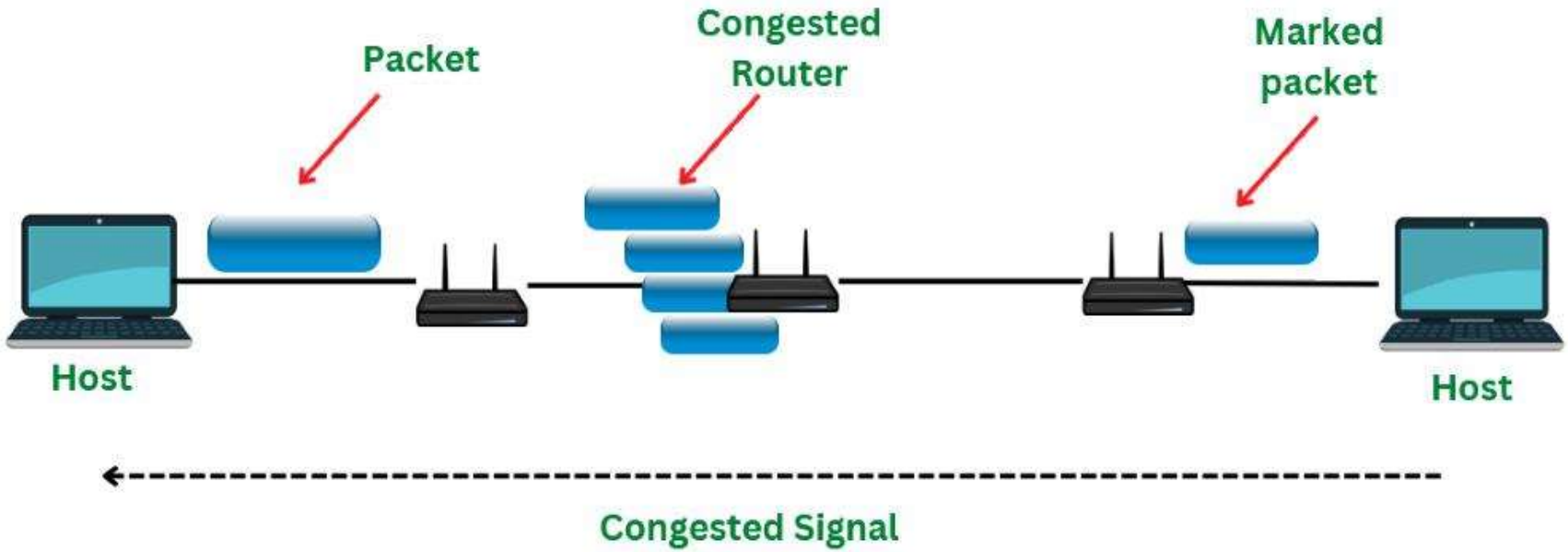
1. Choke packets

- Choke packets are a mechanism where the router directly sends the choked packet back to its sender or host. The header bit of the original packet is turned on so that it will not be able to generate any choke packet. At the time of congestion to decrease the load router will send back only the choked packets at a lower rate. In the case of datagram networks randomly, the packets are selected therefore it leads to more choked packets. The below diagram describes the choke packets approach.


Choke
Packet
I
II
III
IV
Source
Congestion
Destination
Data Flow

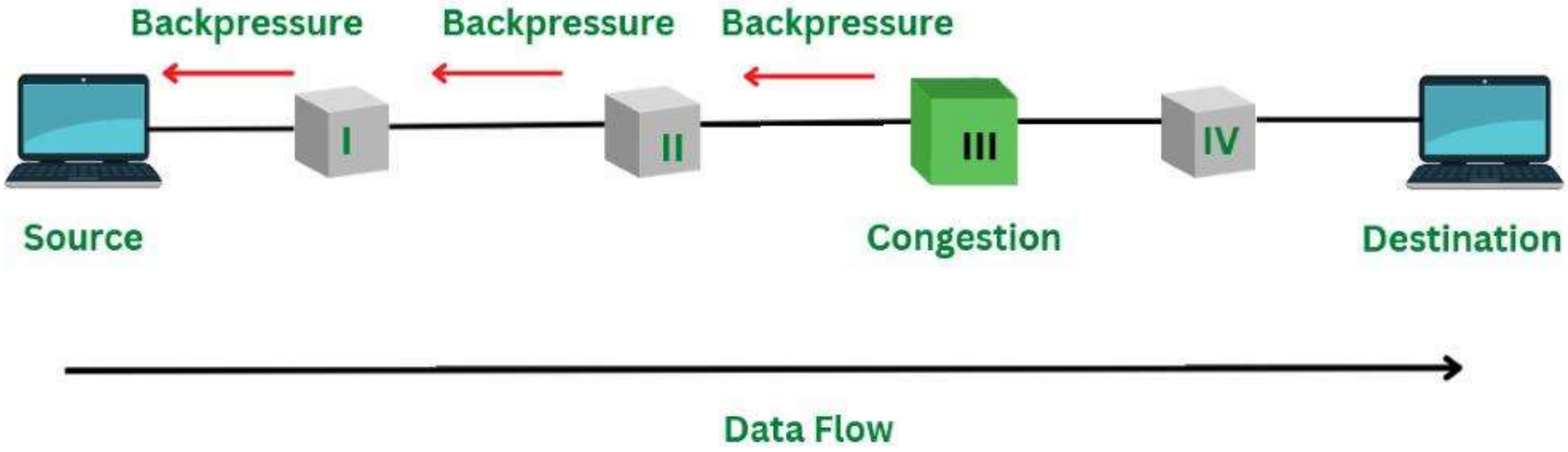
2. Explicit Congestion Notification

- In the explicit congestion notification approach the router does not send extra packets to the host but sets a bit of any one of the packet headers to inform that the network has approached with congestion. When any packet is delivered in the network the destination sends a reply packet to the sender informing that congestion has occurred. In the case of choke packets, the sender then throttles its transmission. The below diagram describes the explicit congestion notification.


Packet
Congested
Router
Marked
packet
Host
Host
Congested Signal

3. Hop-by-Hop Backpressure

- After the congestion has been signaled still due to a slow signal many packets are received from the long distances. The choke packets have an effect at every step and each router requires more buffers. The main aim of this Hop-by-Hop Backpressure technique is to provide faster relief at the point of congestion in the network. This technique propagates in the opposite direction of the data and is majorly used in virtual circuits.


Backpressure
Backpressure
Backpressure
I
II
III
IV
Source
Congestion
Destination
Data Flow

Load Shedding:

- Load shedding is one of the techniques used for congestion control. A network router consists of a buffer. This buffer is used to store the packets and then route them to their destination. Load shedding is defined as an approach of discarding the packets when the buffer is full according to the strategy implemented in the data link layer. The selection of packets to discard is an important task. Many times packets with less importance and old packets are discarded.

Selection of Packets to be Discarded

- In the process of load shedding the packets need to be discarded in order to avoid congestion. Therefore which packet needs to be discarded is a question. Below are the approaches used to discard the packets.

1. Random Selection of packets

- When the router is filled with more packets, the packets are selected randomly for discarding. Discarding the packets it can include old, new, important, priority-based, or less important packets. Random selection of packets can lead to various disadvantages and problems.

2. Selection of packets based on applications

- According to the application, the new packets will be discarded or old packets can be discarded by the router. When the application is regarding file transfer new packets are discarded and when the application is regarding multimedia the old packets are discarded.

3. Selection of packets based on priority

- The source of packets can mark the priority stating how much important the packet is. Depending upon the priority provided by the sender the packet can either be selected or discarded. The priority can be given according to price, algorithm, and methods used, the functions that it will perform, and its effect on another task upon selecting and discarding the packets.

4. Random early detection

- Randomly early detection is an approach in which packets are discarded before the buffer space becomes full. Therefore the situation of congestion is controlled earlier. In this approach, the router initially maintains a specific queue length for the outgoing lines. When this average set line is exceeded it warns for congestion and discards the packets.

QUALITY OF SERVICE:

Techniques for Achieving Good Quality of Service:

Overprovisioning: An easy solution to provide good quality of service is to build a network with enough capacity(buffer space , Bandwidth) for whatever traffic will be thrown at it. The name for this solution is **overprovisioning.**

- The trouble with this solution is that it is expensive.

Buffering

- Flows can be buffered on the receiving side before being delivered. Buffering them does not affect the reliability or bandwidth, and increases the delay, but it smooths out the jitter.
- For audio and video on demand, jitter is the main problem, so this technique helps a lot.

Traffic Shaping:

- Traffic shaping smooths out the traffic on the server side, rather than on the client side and regulates the average *rate (and burstiness) of data transmission*

IMPROVING QUALITY OF SERVICE:

Leaky Bucket Algorithm

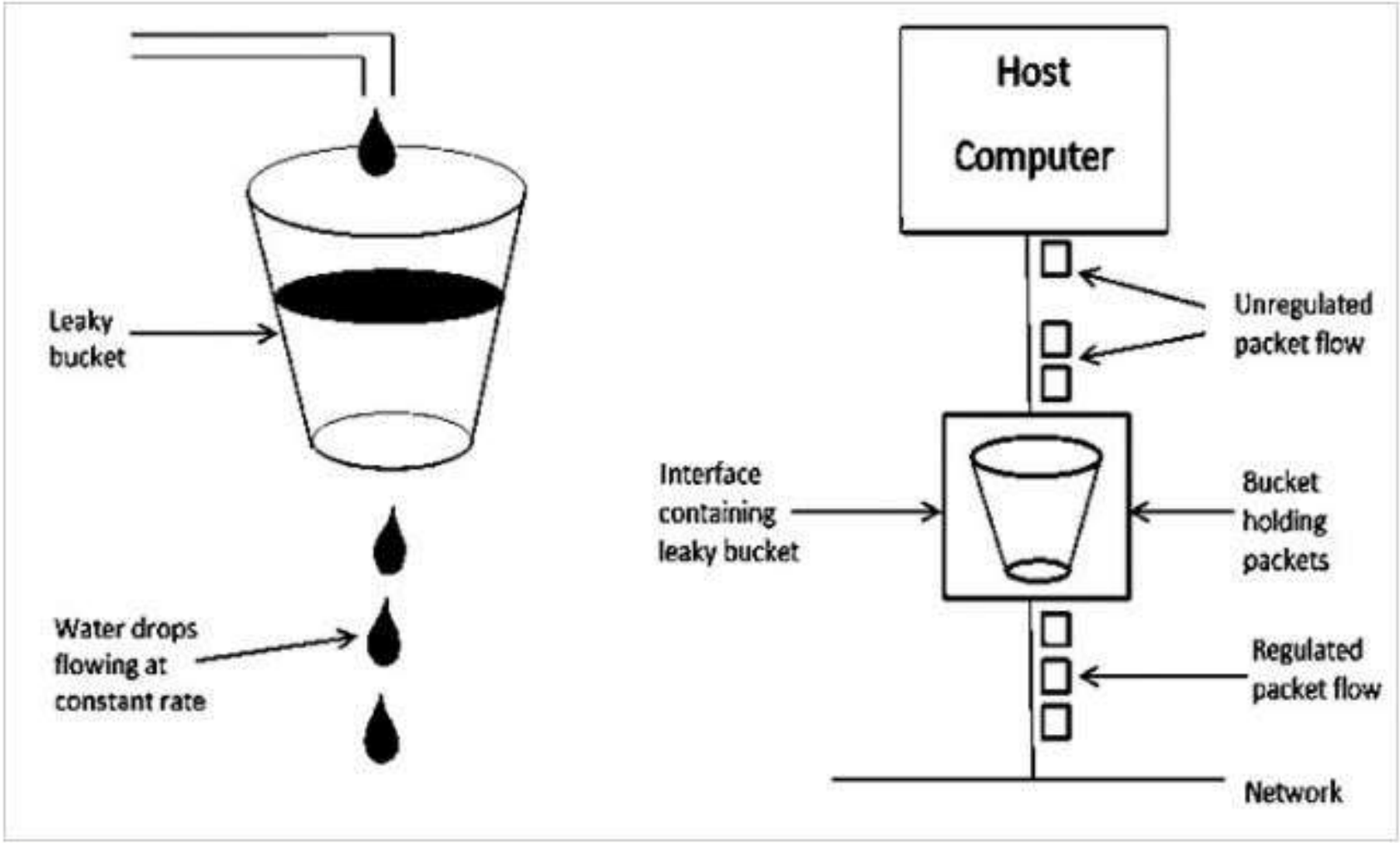
Let us consider an example to understand

- Imagine a bucket with a small hole in the bottom. No matter at what rate water enters the bucket, the outflow is at constant rate. When the bucket is full with water additional water entering spills over the sides and is lost.
- Similarly, each network interface contains a leaky bucket and the following **steps** are involved in leaky bucket algorithm:

1. When host wants to send packet, packet is thrown into the bucket.
2. The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
3. Bursty traffic is converted to a uniform traffic by the leaky bucket.
4. In practice the bucket is a finite queue that outputs at a finite rate.

Leaky Bucket Algorithm

Let see the working condition of Leaky Bucket Algorithm –

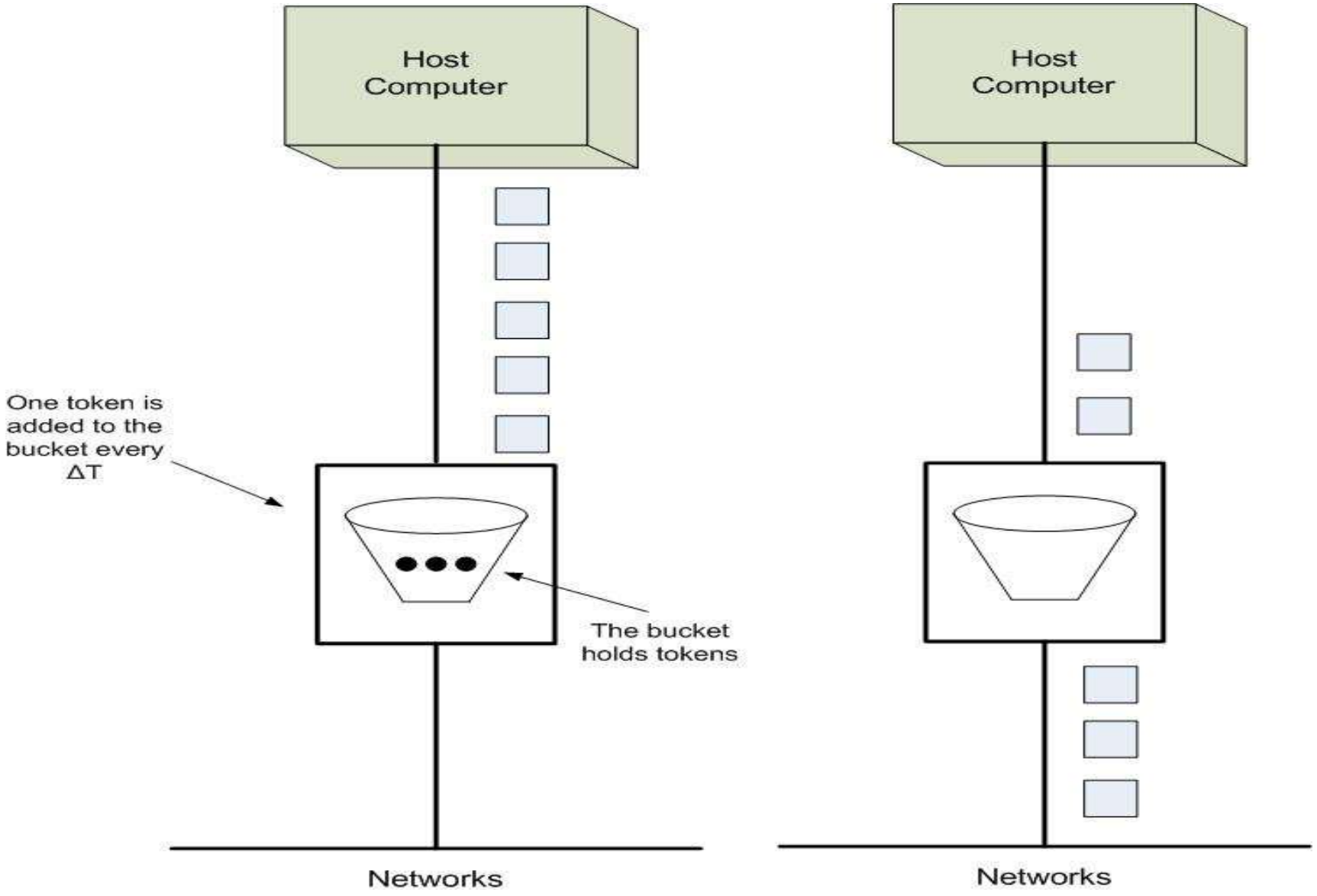


Token bucket Algorithm

- **Need** of token bucket Algorithm:-
- The leaky bucket algorithm enforces output pattern at the average rate, no matter how bursty the traffic is. So in order to deal with the bursty traffic we need a flexible algorithm so that the data is not lost. One such algorithm is token bucket algorithm.
- **Steps** of this algorithm can be described as follows:
- In regular intervals tokens are thrown into the bucket. ƒ
- The bucket has a maximum capacity. ƒ
- If there is a ready packet, a token is removed from the bucket, and the packet is sent.
- If there is no token in the bucket, the packet cannot be sent

- Let’s understand with an example,
- In figure (A) we see a bucket holding three tokens, with five packets waiting to be transmitted. For a packet to be transmitted, it must capture and destroy one token. In figure (B) We see that three of the five packets have gotten through, but the other two are stuck waiting for more tokens to be generated.

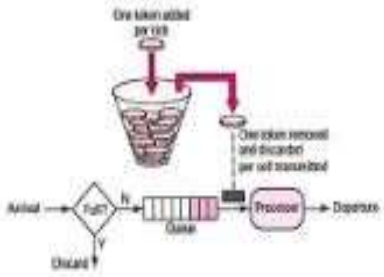
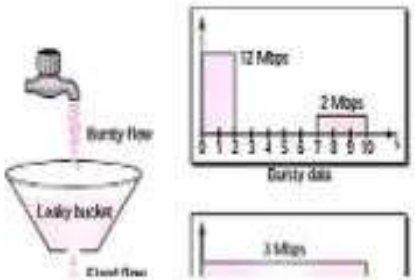
Token Bucket



Ways in which token bucket is superior to leaky bucket:

The leaky bucket algorithm controls the rate at which the packets are introduced in the network, but it is very conservative in nature.

- Some flexibility is introduced in the token bucket algorithm.
- In the token bucket algorithm, tokens are generated at each tick (up to a certain limit).
- For an incoming packet to be transmitted, it must capture a token and the transmission takes place at the same rate.
- Hence some of the busty packets are transmitted at the same rate if tokens are available and thus introduces some amount of flexibility in the system.

Sr no.	Token Bucket	Leaky Bucket
1	Token bucket is token dependent.	Leaky bucket is token independent.
2	 Block diagram token bucket.	 Block diagram of leaky bucket.
3	If bucket is full token are discarded but not the packet.	If bucket is full packet or data is discarded.
4	Token bucket allows for large bursts to be sent faster by speeding up the output.	Leaky bucket sends the packets at an average rate.
5	Token bucket allows saving up of tokens (permission) to send large bursts.	Leaky bucket does not allow saving a constant rate is maintained.
6	Packets can only Transmitted when there are enough token.	Packet are transmitted continuously.
7	It save token.	It is does not save token.