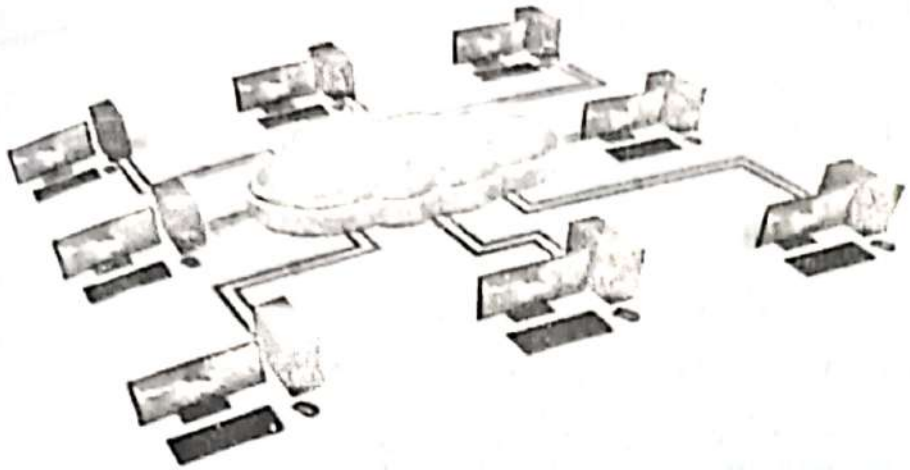


6

NETWORK LAYER

INTRODUCTION

- Network layer is the third layer of OSI model.
- It is the lowest layer which deals with end to end transmission.
- It provides services to the transport layer.
- The primary role of network layer is to route the messages associated with the higher protocol layers above it and also across the network/networks which link the distributed DET' s/systems.
- This layer has a higher responsibility than Data Link Layer because the data link layer is only supposed to move the frame from one end to other.

6.1 BASIC OPERATION OF NETWORK LAYER

- The network layer can operate both in connectionless mode as well as in connection oriented mode.
- In LAN's the frames are addressed and routed between nodes which are attached to the same LAN using their point of attachment address.
- On the other hand a connectionless network layer service and associated protocols is used when all the stations are connected to a single LAN.

- The network layer is also known as inactive layer or Null layer because of its lack of functionality in local area networks.
- If the network comprises a number of inter connected networks instead of a single LAN then network layer protocol is more complex.
- In some of the networks the network layer runs in interface message processors (IMP's or nodes) and the transport layer runs in machine (host) so the boundary between network layer and transport layer also the boundary between subnet and host in these networks.
- We know that the network layer provides its services to transport layer so the network layer services are such designed that there should be no controversy or problems services are as follows :
  - They should be independent of subnet technology.
  - It should sheild the transport layer from type, topology and number of subnets present.
- Before moving into more details about network layer first we should understand what a subnet is subnet is small network in a network.
- Any distributed computing system consists of end systems and of sub network.
- This sub network provides resources for interconnection of the end systems.
- Two types of connections are provided by sub network.
  1. Fixed connection
  2. Switched connection

1. Fixed Connection

- The fixed connection is mostly in form of physical connection media which links the end systems.
- In such networks resources are allocated permanently for use by the end system.

2. Switched Connection

- In the switched connections the concept of selecting and establishing a path from a source to destination through network is used.
- In these networks the resources are shared and allocated temporarily on request of the end systems.

Various sub networks which provide switched connection are

- Switched data sub network
- Circuit switched subnet works
- Message switched sub networks
- Packet switched subnet works

- Now let us discuss the network layer in the layered architecture in brief.
- For the sake of simplicity we will consider the connection oriented services at Physical layer and Data link layer interfaces.
- The network layer provide services to the transport layer at the network layer interface, which interfaces between carrier and customer. The carrier often control the protocols.

Network layer has been designed with the purpose of following goals :

1. **The service should be independent of the subnet technology.**
2. **The transport layer should be shielded from the number, type and topology of the subnet.**
3. **The network addresses made available to the transport layer should use a uniform numbering plan.**

6.2 NETWORK LAYER DESIGN ISSUES

The various issues for the network layer design are :

1. Services provided to the transport layer.
2. Internal organisation of the subnetwork layer.
   - The services provided should be independent of the underlying technology.
   - The transport layer should be shielded from the number, type and different topologies of the subnet user uses *i.e.* all that transport layer wants is a communication link, it need not know how that link is made.
   - With these goals is mind two different types of services emerged.
     1. Connection Oriented Network Services
     2. Connectionless Network Services

1. Connection oriented service is one in which user is given a "reliable" and to end connection.

2. To communicate, the user requests a connection, then uses the connection to their contents, and then closes the connection.
   - A telephone line is best example of connection oriented network.

2. In connectionless, the user simply bundles his information together, put an address on it and then send it off, in the hope that it will reach its destination.

- In this connection there is not guarantor that the message has been reached successfully.
- There is no guarantee that packet has been reached.
- The best example is a letter sent through the post-office.

6.3 SERVICES PROVIDED TO THE TRANSPORT LAYER

The network layer services are designed to achieve the following goals :

1. The service should be independent of the subnet technology.
2. Transport layer should be shielded from the number, type and topology of the subnet.
3. The network address which is made available to the transport layer must use a uniform numbering plan.
   - The network service can be connectionless or connection oriented.
   - The internet has connectionless network layer whereas ATM has connection oriented services.
   - Finally we can say that network layer should provide a raw means to send packets from sender to receiver.

A. Internal Organisation of the Network Layer

- Basically there are two philosophies for organizing the subnet.

1. To use connection oriented services.
2. To use connectionless services.

- In the connection oriented service, a connection is called virtual circuit. It is similar to physical connection.
- In the connectionless organization, the independent packets are called as datagram.

A.1 Virtual Circuit

- The principle behind the virtual circuit is to choose only one route from source to destination.

- When a connection is established, it is used for all traffic flowing over the connection.
- When the connection is released, the virtual circuit is terminated.

Features of Virtual Circuit

- In virtual circuit the router has to maintain the table.
- Each packet must have a virtual circuit number field in its header in addition to the check sum number.
- Circuit set up is required in virtual circuit.

A.2 Datagram

- With a datagram, the routes from source to destination are not worked out in advance.
- Each packet sent is routed independently.
- Successive packet can follow different routes.
- The datagram subnet have to do more work but they are more robust and deals with failures and congestion more easily as compared to virtual circuit.

Features of datagram

- The router do not have to maintain the table.
- Each datagram must contain full address.
- When a packet comes in, the router finds an available outgoing lines and sends the packet out on that line.

Comparison of Virtual Circuit and Datagram subnets.

Sr. No.	Parameter	Virtual Circuit Subnet	Datagram Subnet
1.	Circuit Set up	Required.	Not Required.
2.	Addressing	Each packet contains a short vc number.	It contains source as well as destination address.
3.	Repair	Difficult to Repair.	Easy to repair.
4.	Routing	Fixed. All packet follow the same route. This is static routing	Not fixed, Packet is routed in-dependently.
5.	Congestion Control	Easy.	This is dynamic routing Difficult.
6.	Chance of Failure	Very rare chance of failure.	Failure rate is high.

6.4 FUNCTIONS OF NETWORK LAYER

- The network layer provides its services to transport layer and for that it takes services of data link layer.

- Its functions are carried out by adding a header to every Network service data unit (N-SDU).
- This header is in form of Protocol Control Information (PCI).
- Thus formed Network protocol data unit is transported over existing data link connection. The functions of network layer are as follows.

Network layer is layer 3 of OSI reference model.

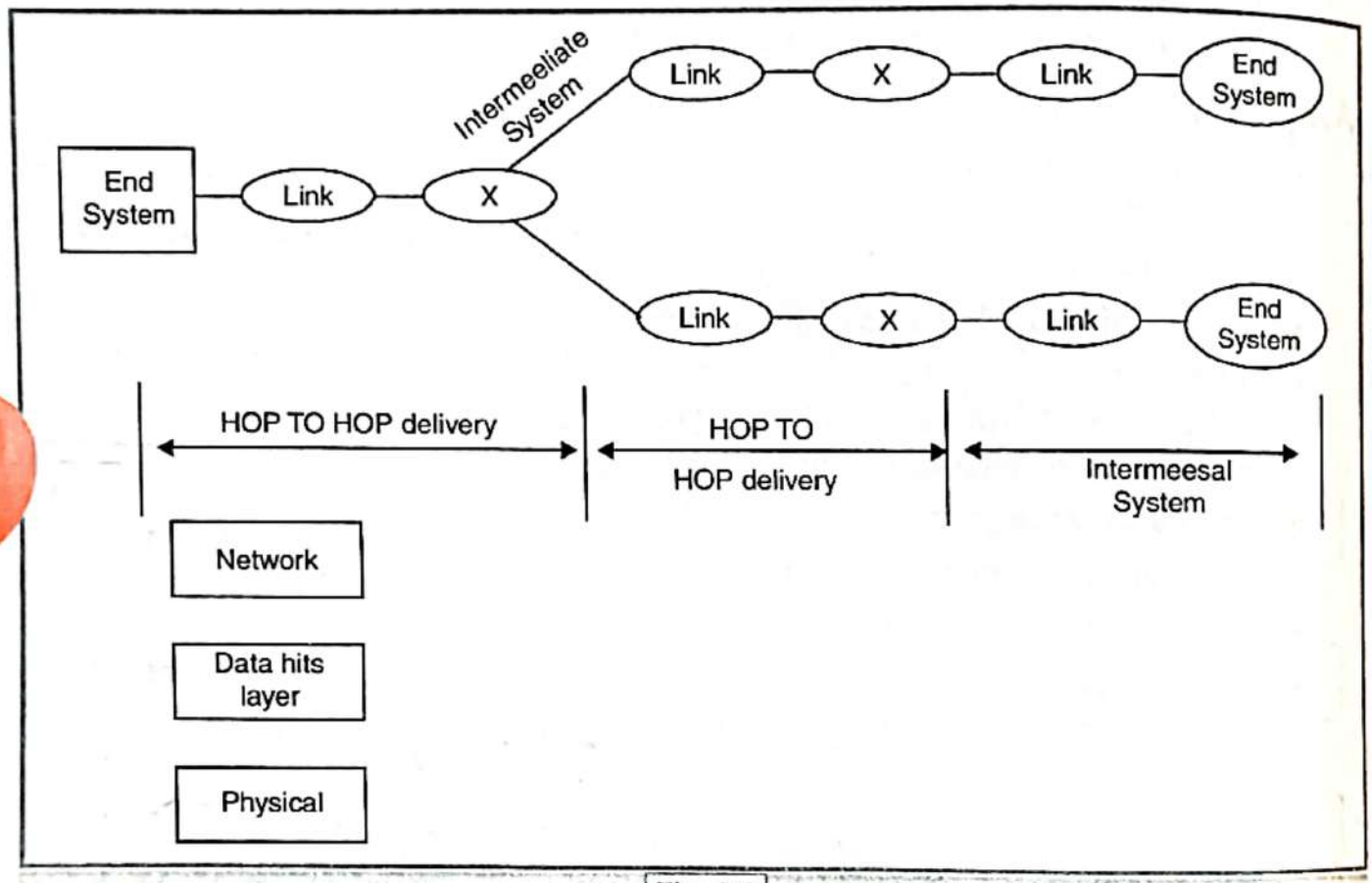


Fig. 6.1

1. Network Connection :

- When network layer receives a call request from transport layer it establishes a network connection across the sub network.
- For this it makes use of data link connections.
- The network layer of the end system interacts with the network layer of the sub network access point for this purpose.

2. Routing :

- The network layer selects an appropriate route between source and destination machines.
- This route is either decided in the beginning or for each N-SDU depending on type of services.

- The routing functions are facilitated by sub layering of the network layer.
- The routing decisions are taken using routing algorithms.
- If the subnet uses virtual circuit, routing decisions are made when the virtual circuit is being set up.
- If the subnet uses datagram's the routing decisions are made a new for every arriving data packet.

3. Multiplexing :

- For optimum use of data link connections, some times many network connections are required to be multiplexed on a single data link connection.
- This is done by network entity which may multiplex serveral network connections on a data link connection. See fig.

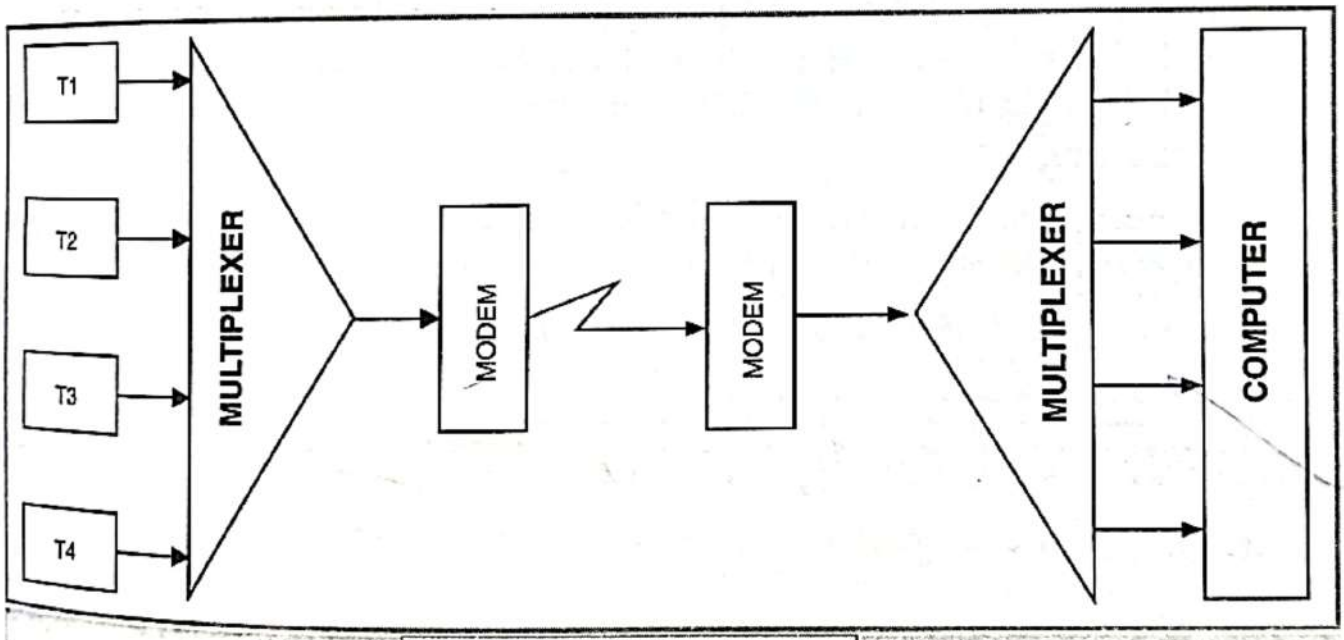


Fig. 6.2. A multiplexed system

4. Error Detection :

- The network layer uses error detection functions to ensure that the quality of service provided over the network connection is maintained.
- Most of the errors are detected and corrected at data link layer.
- The residual errors, if any, are notified by data link layer to network layer.
- Depending on quality of service to be provided, the network layer incorporates the mechanism for error recovery.

5. Internetworking

- To provide the internetworking of between different network.
- Provide the logical connections between different types of network.

6. Packetzing

- It receives the data from upper layer and creates its own packets by encapsulating the packets. This process is known as packetzing.
- It addressing formatting is also formed here.

7. Fragmentation

- Fragmentation means dividing the larger packet into small fragment.
- The basic motive behind the fragmentation is that they can be easily sent to the physical medium.

8. Other Functions :

- At the network layer the network service data unit is segmented and blocking is done to produce a network Protocol data unit (N-PDU). The delimiters of N-SDU are preserved during segmenting and blocking.
- The network entity, also carries out sequencing and flow control of network service data unit on request of transport layer.
- On request of transport layer it may reset the network connection too.

6.5 NETWORK SERVICE TO TRANSPORT LAYER

- The services provided by network layer to transport layer are known as network service.
- The network service provides a transparent transfer of Network Service Data Units (N-SDU's), between two transport layers of end systems, which are received at Network Service Access point (N-SAP) and are delivered at N-SAP.

Two types of network services are

1. Connection Mode Network Service.
2. Connectionless Mode Network Service.

1. Connection Mode Network Service :

- Connection mode network service is a reliable service. It has built in error recovery procedures.

- In this type of service, before transporting data, a network connection is established between communicating end systems transport entities.
- In case of network connection failure transport entities are informed by the network entities.

2. Connectionless Mode Network Service

- Connectionless mode network service is less reliable and each N-SDU in it contains address of source and destination end systems.
- The routing decisions are taken by network layer of nodes and routes can be different of different N-SDU.
- In these services out of sequence delivery of N-SDU's, duplication and loss of N-SDU's can take place.
- The transport entities have to make their own efforts to correct the delivery of N-SDU's.
- The network service can be of any of the above two types but their basic features are almost the same except a few. The main features of network services are as follows :

Features

1. It does not restrict coding or content format of user data so it is transparent.
2. It contains functions necessary to make differences in the characteristics of different subnet works.
3. It takes care of special requirements, for formatting of data, for different sub networks.
4. The unique identity of networks i.e. network addresses are known to it.
5. It establishes, maintains and releases the network connection on request of service users and/or transport layer.
6. The quality of service is maintained during the connection. Some times if they are unable to do so they inform about it to transport entities.
7. It notifies unrecoverable errors to transport entities.
8. It provides flow control across the layer interface.
9. It provides sequence delivery of network service data units over a network connection on request of transport layer.
10. It can reset the network connection and it is an optional feature.

11. Expedited data transfer and receipt of data over a network connections can also be provided. Both of them are optional features.

6.6 SERVICE PRIMITIVE :

- Various services offered by network entities are Connect, Data, Data-Ack, reset, disconnect, unit data, expedited data and facility. Various network service primitives along with service names.
- These primitives are defined by international standard 8348 for OSI connected oriented network services. The service primitives are for connectionless network.
- The four categories of connection oriented service primitives are establishing (CONNECT), releasing (DISCONNECT), Using (DATA, DATA-ACKNOWLEDGE,(Ex. PEDITED-DATA) and resettling (RESET).
- Most of them have parameters and the method of parameter passing in implementation dependent.
- Various parameters are callee, caller, QOS, user data etc.
- The connectionless service primitives have three categories Unit data, Facility and Report. The way of use of these primitives are not defined in the standard.
- They are thus network dependent.
- The network layer also provides a uniform naming for the transport layer to use.
- All the network service primitives use NSAP addresses for identifying source and destination.
- Thus NSAP address has three fields.
- The first one is Authority and format identifier (AFI) which identifies the type of address present in the third field known as Domain specific part (DSP).
- The second field is internal domain identifier (IDI).
- The AFI and IDI are jointly known as internal domain part (IDP).

6.7 ROUTING ALGORITHMS

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

- In most subnets, packets will require multiple hops to make the journey.
- The routing algorithm is the part of network layer software responsible for deciding which output line an incoming packet should be transmitted.
- If the subnet uses datagram externally this decision must be made for every arriving data packet since the best route may hve changed since last time.
- If the subnet uses virtual circuit externally, routing decision are made only when a new virtual circuit is being set up.
- There are certain properties that are desirable in the routing algorithm are (1) correctness, (2) simplicity, (3) robustness, (4) stability, (5) fairness (6) optimility.
- The routing algorithm should be able to cope with changes in the topology and traffic.
- Stability is also an important goal for the routing algorithm.
- Routing algorithm can be grouped into two major classes : **nonadaptive and adaptive**.
- Nonadaptive Algorithm do not base their routing decision on measurement or estimates of the current traffic and topology. This procedure some times called **static routing.**
- Adaptive algorithm change their routing decision to reflect the change in topology.
- Adaptive algorithm differ in where they get number of hops or estimate transmit time.
- The set of optimal routes from all sources to a given destination form a tree rooted at the destination, such tree is called a **sink tree**, where distance metric is number of hops.
- A sink tree is indeed a tree it does not contain any loop.
- So each packet will be delivered with in a finite and bounded number of hops.
- A routing algorithm must be able to cope with changes in the topology and traffic without rebooting the network.
- Correctness and simplicity is no doubt required.
- Stability is also important. Fairness and optimality are required to improve delay and reduce amount of band width consumed.

There are two major classes of routing algorithms

1. Adaptive Algorithm
2. Non adaptive Algorithm

6.7.1 Adaptive Algorithm/Dynamic

- The adaptive algorithms base their routing decisions on measurements or estimates of the current traffic and topology.
- Adaptive algorithms are divided into three parts.
  * Isolated
  * Centralized
  * Distributed

Features

1. These algorithms are dynamic.
2. Routing decisions to reflect changes in the topology traffic.
3. Continuously update their information about the network and routing decisions are based on the current state of network.

6.7.2. Non-Adaptive/Static

- In the non adaptive algorithms the choice of route is computed in advance. The distance is computed off line and down loaded to the IMP's when the network is booted.
- This process is known as static routing. The two techniques mostly used in routing algorithms are
  - Shortest path routing
  - Multi path routing or Bifurcated routing.

Features

1. In this algorithm, routing decisions are not based on the measurement or estimation of current traffic and topology.
2. The route from a particular source to destination is calculated in advance and is downloaded to the router, when the network is initialized.
3. These type of algorithm is called static algorithm.

6.8 MORE ROUTING ALGORITHM

6.8.1 Shortest path routing :

- In this technique to choose a route the algorithm finds shortest path between two nodes.

- For measuring path length it can use number of hops, geographical distance in kilometers or labeling of arcs.
- The labeling of arcs can be done with mean queuing and transmissions delay for a standard test packet on hourly basis or daily basis or can be computed as a function of bandwidth, distance, average traffic, communication cost, mean queue length, measured delay or some other factors.
- There are various algorithms which compute shortest path between two nodes of a graph. For example Dijkstra which uses labeling of nodes with its distance from source node along the best known path.
- Initially all nodes are labeled with infinity and as algorithm proceeds the label may change. The labeling for graph in fig.

Can be done in various passes as follows

Pass 1. B (2,A), C (¥, –), F (¥, –), e (¥, –), d (¥, –), G (6, A)

Pass 2. B (2, A), C (4, B), D (5, B), E (4, B), F (¥, –), G (¥, –)

Pass 3. B (2, A), C (4, B), D (5, B), E (4, B), F (7, C), G (9, D)

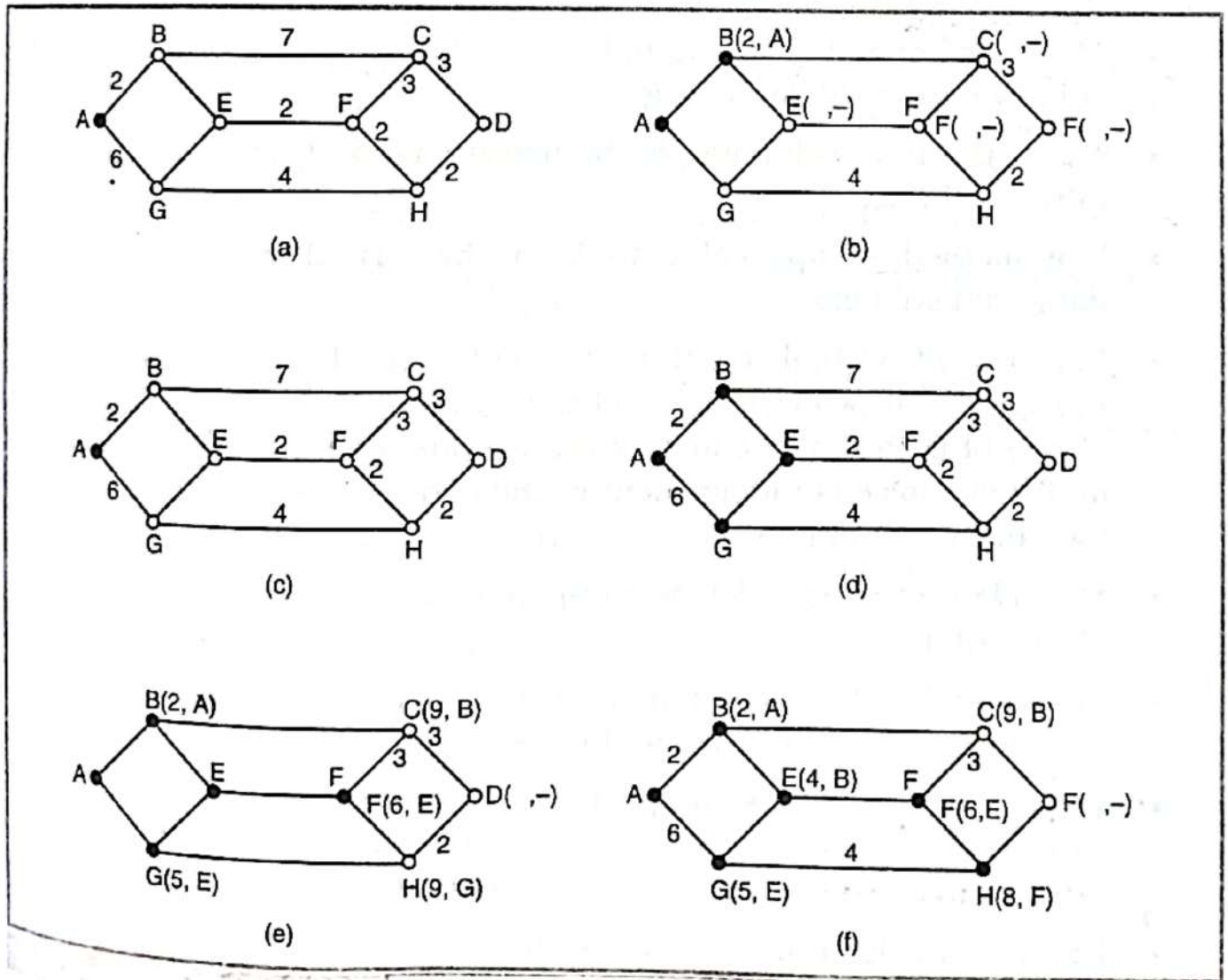


Fig. 6.3 Shortest Path Routing

- We can see that there can be two paths between A and G.
- One through ABCFG and other through ABDG.
- The first one has path length as 11 while second one has 9.
- Hence second one as G (9, D) is selected.
- Similarly Node D has also three paths as ABD, ABCD and ABED.
- First one has path length as 5 rest two have 6.
- So first one is selected.
- Actually all nodes are searched in various passes and finally the routes with shortest path lengths are made permanent and the nodes of the path are used as working node for the next round.

6.8.2 Multipath Routing :

- Sometimes in a network it is found that there are more than one path with shortest length between pairs of nodes.
- In such cases to reduce the load on each of communication lines, the traffic is splitted up over several paths.
- This technique of using multiple routes between a single pair of nodes is known as multiple routing.
- As it used a technique of bifurrication so it is also known as bifurcated routing.
- The method is applicable to both the virtual circuit as well as datagram subnets.
- In case of virtual circuits different virtual circuits are routed independently while in case of datagram subnets at each node the choice of path is made among the various alternatives for the packet and this choice is independent of the choices made for other packets for same destination.
- Multiple routing is widely used to improve performance and reliability of the subnet.
- Routing tables are maintained at each node which contains choices with destination node, line and respective probability.
- When node receives a packet for a certain destination. The corresponding row is selected and then node generates a random number to decide which line should be selected.
- Let us consider the example of graph subnet. The routing table for node D is shown in fig. 6.2.

- If the node D receives a packet whose destination is E it uses row number 5 (row labeled with E) for deciding the route.
- It then generates a random number between 0.00 and 0.99. If number is less than 0.70 the line E is used. If it is between.70 and 0.90 the line A is used else line B is used.

The main advantage of bifurcated routing lies in the possibility of sending different types/classes of traffic over different paths.

Now let us discuss three types of adaptive algorithm in more detail.

6.8.3 Isolated Routing :

- Those algorithms which run separately on each Interface Message Processor (IMP) or network layer and only use information available they are called isolated adaptive routing algorithms.
- These algorithms try to adapt to changes in topology and traffic.
- **Baran's hot potato algorithm, backward learning algorithm and flooding are** examples of isolated routing algorithms.

6.8.3.1 Hot Patato Algorithm

- In **hot potato** algorithm when a packet comes, the network layer of the node tries to get rid of it as soon as possible and for that it puts the packet on shortest output queue.
- For selecting a queue only its length is considered.
- In a variation of hot potato algorithm both queue length and static weight of lines are considered.
- In such algorithms either shortest route is selected with low static weight or with best static weight choice a queue is selected with a threshold value.

6.8.3.2 Backward learning Algorithm

- In backward learning a counter is used with each packet which is incremented on each hop.
- The packets contain address of source machine too.
- Each node knows the sender as well as the number of hops to reach there.
- Each node looks for the best route up to source and compares it from that of packet and thus the shortest path is discovered for marking the route.

6.8.3.3 Flooding Algorithm

- In this routing algorithm each incoming packet is sent out on every outgoing line except for the one it arrived on.
- It generates, thus, large number of duplicate packets.
- To reduce this a counter is used with each packet which is decremented at each hop and a one whose counter value is zero is discarded.
- Most of the times the sender knows the path length of destination but if in case it does not know the counter in initialized by a number equal to full diameter of the subnet.
- Flooding is not practical in most of the applications.
- A more practical variation is called as selective flooding. Random walk algorithm is also a variation of it which is non adaptive algorithm.

6.8.4. Centralized Routing :

- In centralized routing a Routing Control Centre (RCC) is required in the network.
- Every node sends status information to RCC periodically.
- The routing control centre collects information about network from all nodes and then bases upon these information the optimal routes are decided using techniques like shortest path routing. This method has various drawbacks.
  * For large networks the routing calculations take much time.
  * For changing traffic the routing calculations are to be performed each time a change occurs.
  * If routing control center fails the entire network fails.
  * Distribution of routing tables concept sometimes leads to inconsistency because the machines which are closer to routing control center get their new tables first, when ever a change occurs, while those which are far off receive their tables later on causing inconsistency and delay in packets. Of course the delayed packets also contains new routing tables for distant machines causing a more delay added to further packets.
  * The routing control center is heavily loaded as each machine reports to it and it is the one which decides routes for all packets.

* In case the RCC computes optimal route with no alternates for each pair of nodes, loss of even a single line may cut off some machines from the subnet.

TYMNET is an example of centralized routing.

6.8.5 Distributed routing :

- In this class of routing algorithms the routing information of each node is exchanged periodically with neighbors.
- A routing table is maintained at each node with only one entry for each other node of the subnet.
- The table is indexed on this node entry which has two parts
  (a) Outgoing lines from a node
  (b) Distance for destination node via that line
- Each node knows the distance of each of its neighbors.
- The distance can be in terms of hops, delays or queue lengths.
- Each node sends the routing table computed by it to all other nodes and thus each node knows about the distance between any two nodes. Now each node has complete information about distances of subnet nodes which is exchanged periodically.
- Once new routing table is received by a node the old one is rejected automatically.
- These old routing tables are not used in the calculations.
- For computing route for node the source node makes use of the claims depending on information exchanged by other nodes.
- For example if a node X knows that it can get to A in 10 seconds and A has a claim to reach B in 15 seconds then through A it computes the distance as 25 seconds.
- Similarly it computes the route up to B through each node according to information given by them.
- Finally it selects the shortest route in terms of seconds and that route is marked for use.

6.8.6. Flooding :

- Flooding is another static routing algorithm, in which every incoming packet is sent out on every outgoing line except the one it arrived on.
- Flooding creates vast number of duplicate packets.

- One measure to damp the process of an indefinite number is to have a hop counter contained in the header of each packet, which decremented at each hop with the packet being discarded when the counter becomes zero.
- The hop counter should be intialized to the length of the path from source to destination.
- An alternative technique for damming the flood is to keep track of which packet have been flooded to avoid sending them out a second time.
- One way to achieve this goal is to have source router put a sequence number in each packet it receives from its hosts.
- Each router then needs a list per source routee telling which sequence number originating at that source have already been seen.
- If an incoming packet is not the list, it is not flooded.
- A variation of flooding that is slightly more practical is selective flooding. In this type the routers do not send every incoming packet out on every line, only on those line that are going approximately in the right direction.
- Flooding is not practical in most applications but it is useful where large number of routers are used and tremendous robustness of flooding is highly desirable.
- In distributed database applications, it is some time necessary to update all the database concurrently, in which flooding can be useful.

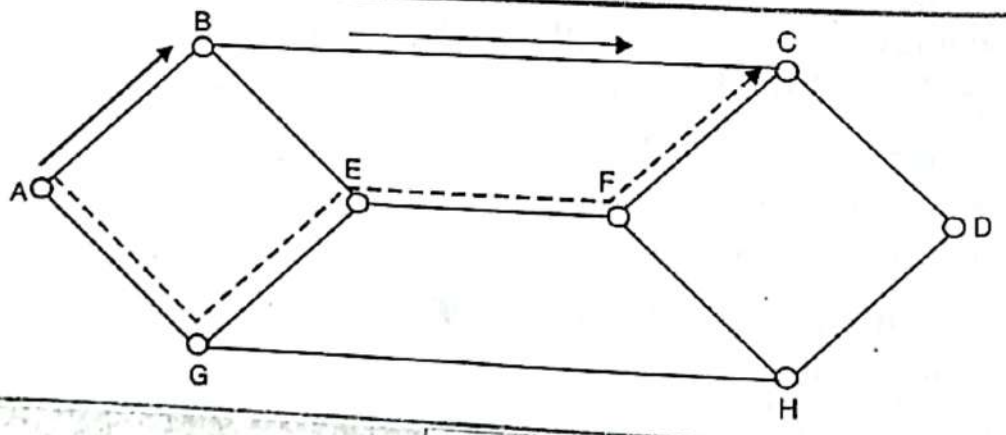


Fig. 6.4

- It is possible to optimise the routing by analysing the data flow mathematically. This is possible if the average traffic from one node to other is known in advance and it is constant in time.
- The mathematical analysis is based on idea that for a given line If the capacity and the average data flow are known, then it is possible to calculate the mean packet delay using queuing theory.

- From the mean delay on all the lines it is possible to calculate the mean packet delay for the whole subnet.
- To use the technique of flow based routing the following information should be known in advance :

1. Subnet Topology
2. Traffic Matrix
3. Line Capacity matrix which specifies capacity of each line.

6.8.7. Flow-Based Routing

- Flow based routing is static algorithm that uses both topology and load for routing for deciding a route.
- The basic idea behind the analyses is that for a given line, if the capacity and the average flow are known it is possible to compute the mean packet delay on that line from queuing theory.
- To use this technique, certain information must be known in advance. First the subnet topology must be known. Second the traffic matrix. Third the line capacity matrix.

6.8.8. Dynamic Routing Algorithms

- In modern computer networks normally dynamic routing algorithms are used.
- There are two dynamic routing algorithm namely distance vector algorithm and link state routing are popular.
- Both these algorithms are suitable for packet switched network.
- Both these algorithms assume that router knows the address of each neighbour and the cost of reaching each neighbour.
- In distance routing, a node tell its neighbour about its distance to every other node in the network.
- Where as in the link state routing, a node tells every other node in the network its distance to its neighbours.
- Hence both the algorithms are suitable for large inter networks.

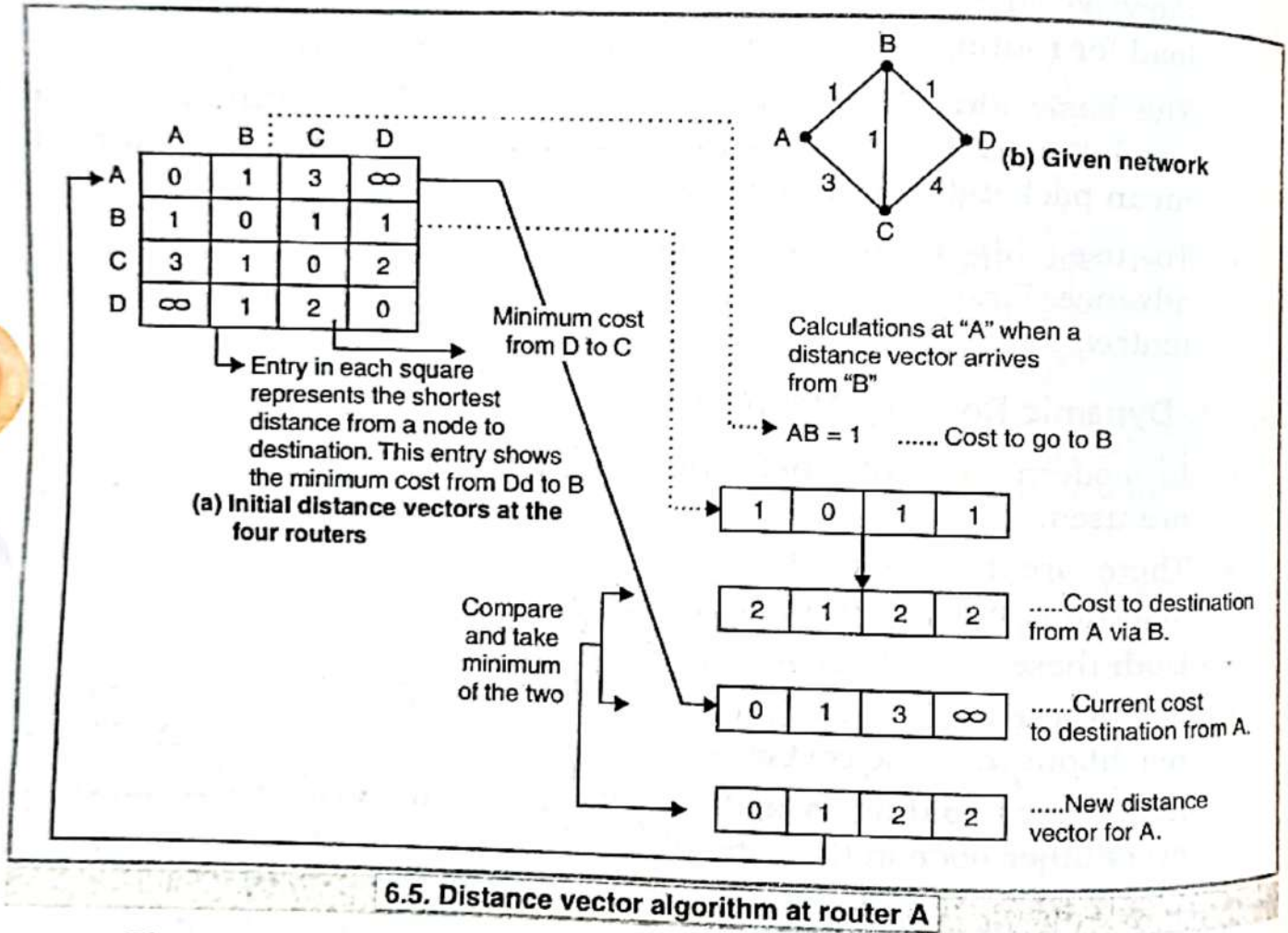
6.8.9 Distance vector :

- In distance vector routing, we assume that each router knows the identity of every other router in the network, but the shortest part to each router is not known.
- A **distance vector** is defined as the list of <destination, cost> tuples, on tuple per destination. Each router maintains a distance vector.

- The cost in each tuple is equal the sum of costs on the shortest path to the destination.

Updation of router tables :

- A router periodically sends a copy of its distance vector to all its neighbours.
- When a router receives a distance vector from its neighbour, it determines whether its cost to reach any destination would decrease if it routed packets to that destination through that neighbour. This is illustrated in Fig. 6.5.



6.5. Distance vector algorithm at router A

- Fig. 6.5 shows how the D.V. at A is automatically modified when a D.V. arrives from B.
- A similar calculation takes place at the other routers as well. In Fig. 6.5 (a) initial distance vector is shown. The entries correspond to the costs corresponding to the shortest distance between the routers corresponding to that square.
- For example, AC = 3 indicates the cost corresponding to the shortest path in terms of number of hops from A to C.

- Even if nodes asynchronously update their distance vectors the routing tables eventually converge.
- The well known example of distance vector routing is the Bellman-Ford algorithm.

Routing procedure in distance vector routing :

- The example of a subnet is shown in Fig. 6.5(A) and the routing tables are shown in Fig. 6.5.

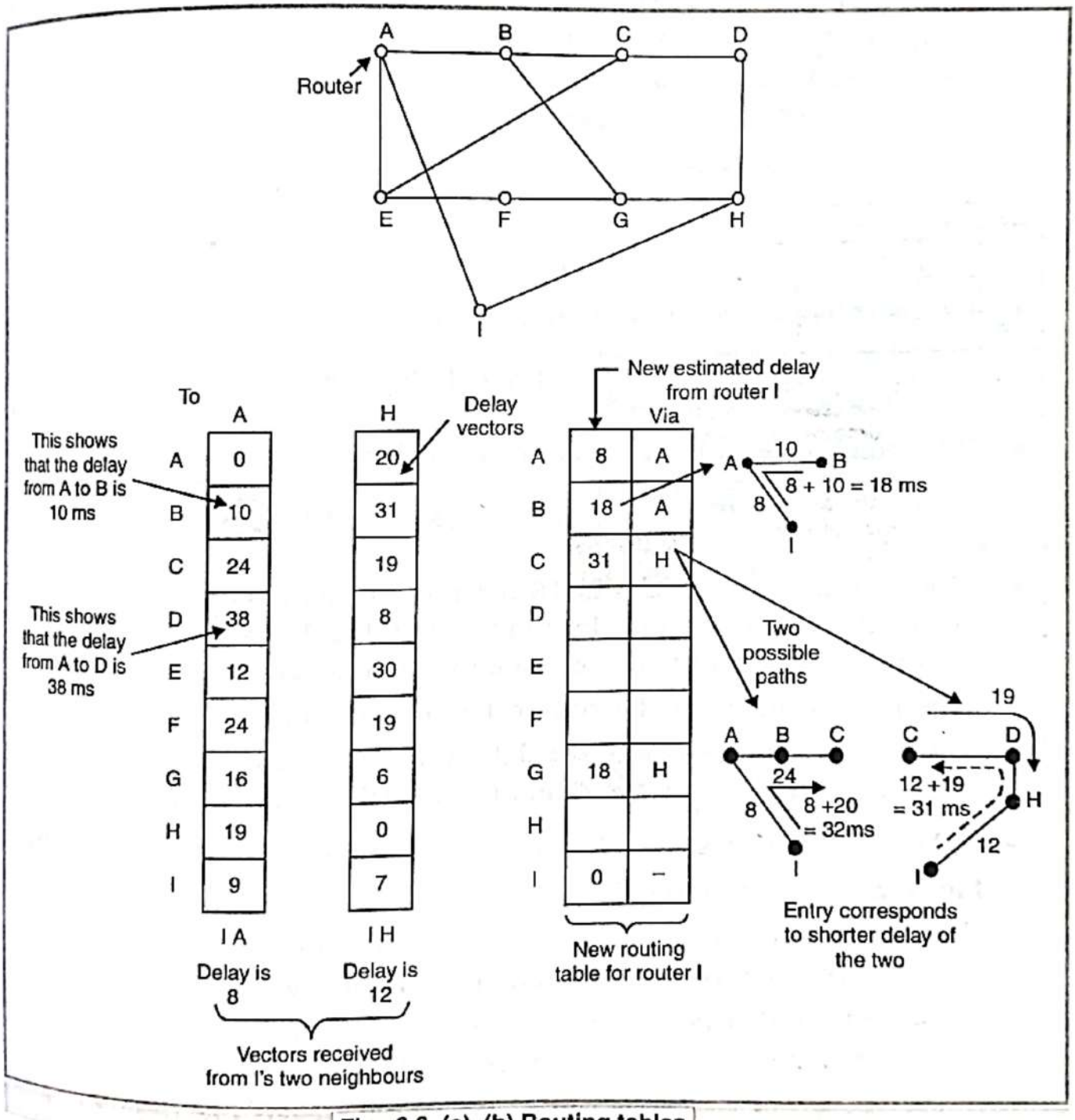


Fig. 6.6. (a) (b) Routing tables

- The entries in router tables of Fig. 6.6 (b) are the delay vectors. For example consider the shaded boxes of Fig. 6.6(b).

- The entry in the first shaded box shows that the delay from A to B is 10 msec, whereas the entry in the other shaded box indicates that the delay from A to D is 38 msec.
- Consider how router I computes its new route to router G. Fig. 6.6 (c) shows the two possible routes between I and G.

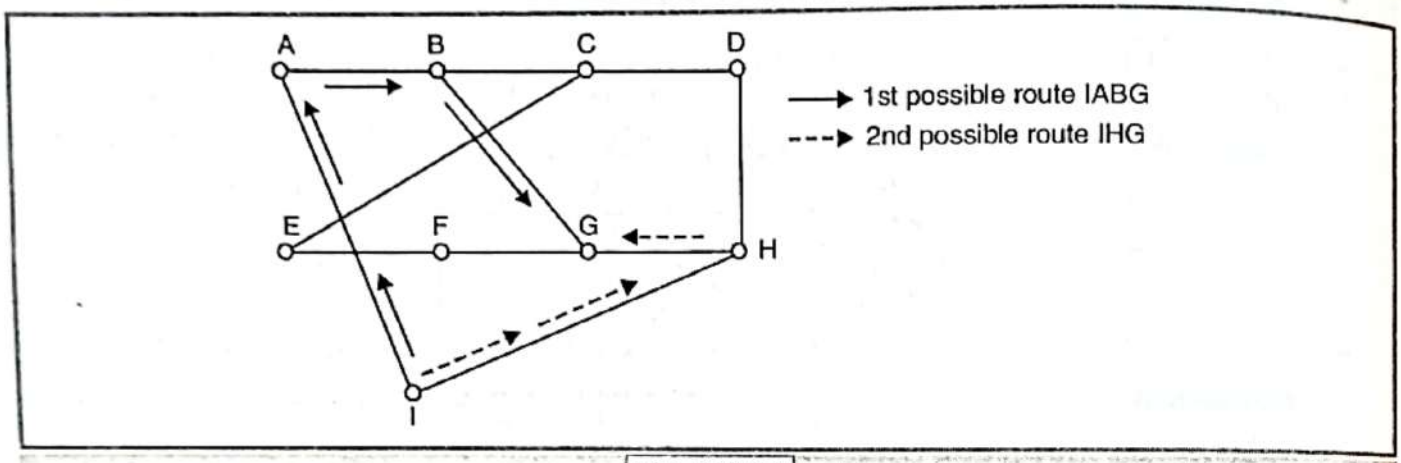


Fig. 6.6 (c)

- I knows that the reach G via A, the delay required is :

$$\left.\begin{array}{ll} \text{I to A} & \text{Delay} = 8 \text{ ms} \\ \text{A to G} & \text{Delay} = 16 \text{ ms} \end{array}\right\} \therefore \text{I to G Delay} = 8 + 16 = 24 \text{ msec}$$

- Whereas the delay between I and G via H (route IHG) is :

$$\left.\begin{array}{ll} \text{I to H} & \text{Delay} = 12 \text{ ms} \\ \text{H to G} & \text{Delay} = 6 \text{ ms} \end{array}\right\} \therefore \text{I to G Delay} = 12 + 6 = 18 \text{ msec}$$

- The best of these values is 18 msec corresponding to the path IHG. Hence it makes an entry in its routing table (I's table) that the delay to G is 18 msec and that the route to use it is via H.
- The new routing table for router I is shown in Fig. 6.5(b).
- Similarly we can calculate the delays, from I to different destinations from A to I and enter the minimum possible delay into the I's router table.

Disadvantages :

1. The problem with distance vector routing is its slowness in converging to the correct answer. This is due to a problem called count-to-infinity problem.
2. This problem can be solved by using the split horizon algorithm.
3. Another problem is that this algorithm does not take the line bandwidth into consideration when choosing root.
4. This is a serious problem which led to the replacement of this algorithm by the Link State Routing algorithm.

6.8.10 Count to Infinity Problem :

- The distance vector routing works properly theoretically but practically it has a serious problem. Although we get a correct answer, we get it slowly.
- In other words it reacts quickly to good news but does so leisurely to bad news.
- Consider a router whose best route to destination X is large. If on the next exchange neighbour A suddenly reports a short delay to X, the router will switch over and start using the line to A for sending the traffic to destination X.
- Thus in one vector exchange, it good news is processed.
- Let us see how fast does a good news propagate. Consider a linear subnet of Fig. 6.7 which has five nodes. The delay metric used is the number of hops.
- Assume that A is initially down and that all the other routers know this. So all the routers have recorded that the delay to A is infinity.

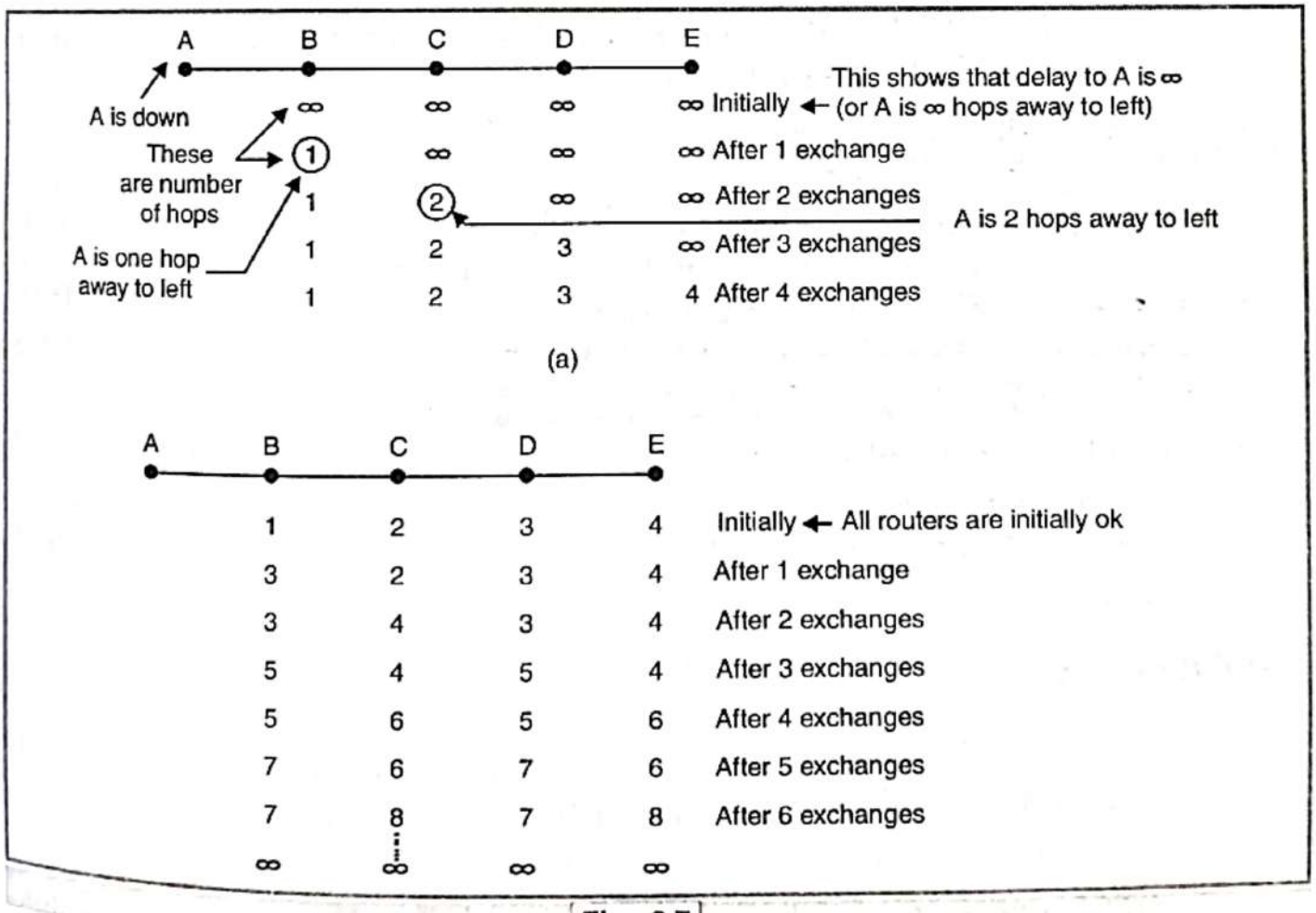


Fig. 6.7

- When A becomes OK, the other routers come to know about it via the vector exchanges. Then suddenly a vector exchange at all the routers will take place simultaneously.

- At the time of first vector exchange, B comes to know that its left neighbour has a zero delay to A. So as shown in Fig. 6.7(a). B makes an entry in its routing table that A is one hop away to the left.
- Al the other routers still think that A is down. So in the second row of Fig. 6.7(a), the entries below C D E are ∞.
- On the second vector exchange, C comes to know that B has a path of 1 hop length to A, so C updates its routing table and indicates a path of 2 hop length. But D and E do not change their table entries.
- So after the second vector exchange the entries in the third row Fig. 6.7(a) are :

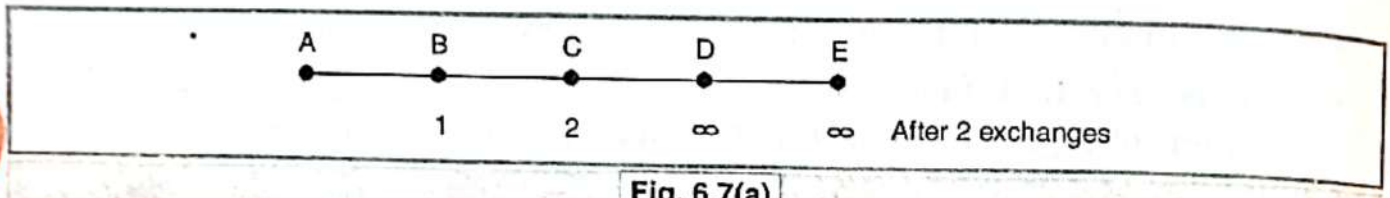


Fig. 6.7(a)

- Similarly D and E will update their routing tables after 3 and 4 exchanges respectively.
- So we conclude that the good news of A has recovered has spread at a rate of one hop per exchange.

Explanation of Fig. 6.7(b)

- Now refer Fig. 6.7(b). Here initially all routers are OK. The routers B, C, D and E have distances of 1, 2, 3 and 4 respectively to A. So the first row of Fig. 6.7(b) is as follows :

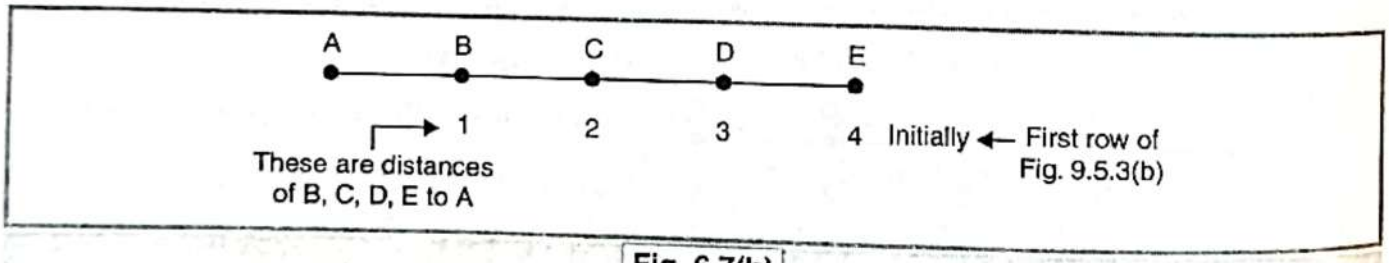


Fig. 6.7(b)

- Now imagine that suddenly A goes down or line between A and B is cut.
- At the first packet exchange B does not hear anything from A (because A is down). But C says "I have a path of length 2 to A". But poor B does not understand that this path is through B itself.
- So B thinks that it can reach A via C with a path length 3. (B to C 1 hop and C to A 2 hops) so it accordingly updates its routing table. But D and E do not update their entries. So the second row of Fig. 6.7(b) looks as follows :

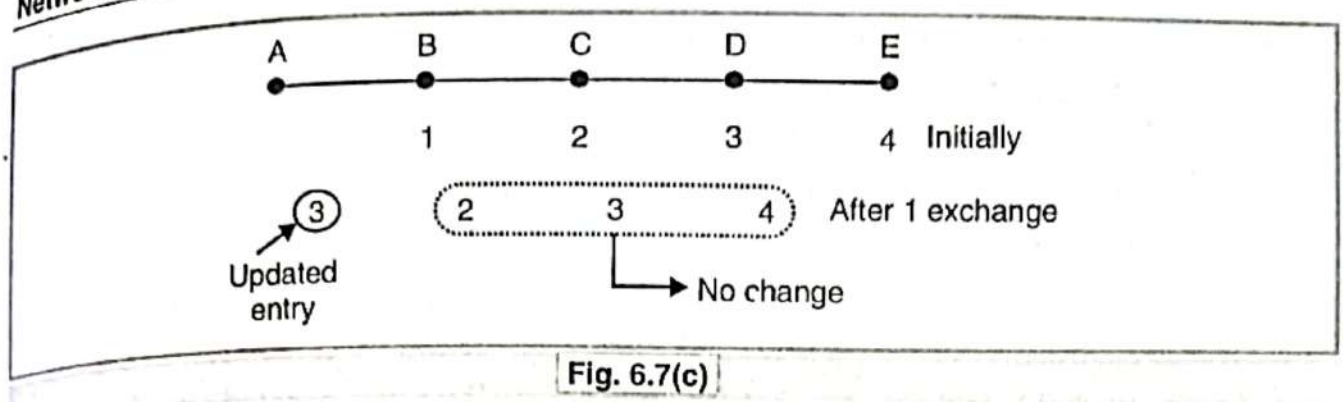


Fig. 6.7(c)

- On the second exchange C realizes that both its neighbours (B and D) claim to have a path of length 3 to A. So it picks one ofn in row 3 of Fig. 6.7(b). It is respected below.

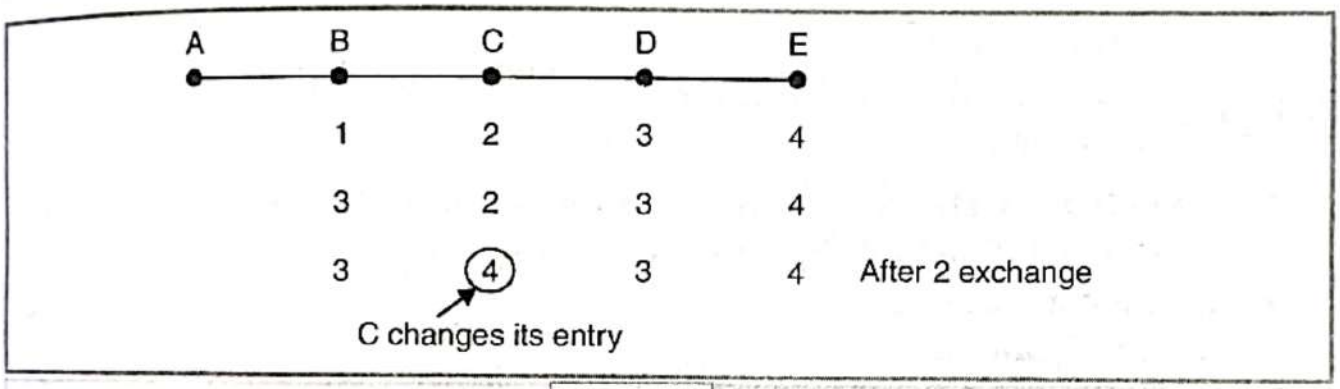


Fig. 6.7(d)

- Similarly the other routers keep updating their tables after every exchange.
- It is expected that finally we should get ∞ in the router tables of B, C, D and E indicating that A is down. We do reach this state at the end in Fig. 6.7. (b) but after a very long time.
- The conclusion is bad news propagates slowly. This problem is called as **count-to-infinity** problem.
- The solution to this problem is to use the split horizon algorithm.

6.8.10.1 The Split Horizon Algorithm :

- This algorithm works the same way as distance vector routing, except that the distance to X is not reported on the line that packets for X are sent on.

6.8.10.2 Link State Routing :

- Distance vector routing was used in ARPANET upto 1979. After that it was replaced by the link state routing.
- Variants of this algorithm are now widely used.

- The link state routing is simple and each router has to perform the following five operations.

Router Operations :

1. Each router should discover its neighbours and obtain their network addresses.
2. Then it should measure the delay or cost to each of these neighbours.
3. It should construct a packet containing the network addresses and the delays of all the neighbours.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.
   - The complete topology and all the delays are experimentally measured and this information is conveyed to each and every router.
   - Then a shortest path algorithm such as Dijktra's algorithm can be used to find the shortest path to every other router.

Protocols :

- Link state routing is popularly used in practice.
- The OSPF protocol which is used in the Internet uses the link state algorithm.
- **IS–IS Intermediate system –** Intermediate system is the other protocol which uses the link state algorithm.
- IS-IS is used in Internet backbones and in some digital cellular systems such as CDPD.

Comparison of Link State Routing and Distance Vector Routing

Sr. No.	Distance Vector Routing	Link State Routing
1.	Each router maintains routing table indexed by and containing one entry for each router in the subnet.	It is the advanced version of distance vector routing.
2.	Algorithm took too long to converge.	Algorithm is Faster.
3.	Bandwidth is less.	Wide bandwidth is available.
4.	Router measure delay directly with special ECHO packets.	All delays measured and distributed to every router.
5.	It doesn't take line bandwidth into account when choosing the routes.	It considers the line bandwidth into account when choosing the routes.

6.8.11 Routing for Mobile Host

- Millions of people have portable computers and they generally want to read their mail and access their normal file system from any location in the world.
- These mobile hosts introduce a new complication to route a packet to a mobile host users who never move are said to be stationary.
- They are connected to the network by the copper wire or fiber optics.
- Migratory users are basically stationary users who move from one fixed site to another from time to time but use the network only when they are physically connected to it.
- Roaming users actually compute on the run and want to maintain their connections are said to mobile users.
- When a packet is sent to a mobile user, it is routed to the user's home LAN. Packet sent to the mobile user on its home LAN are intercepted by the home agent.
- The home agent then looks up the mobile users the foreign agent handling the mobile users. New location and finds the address of the home agent then does two things.
- First, it encapsulates the packet in the pay load field on the outer packet and sends the latter to the foreign agent. This mechanism is called tunneling.
- Second, the home agent tells the sender to send packet to the mobile host by encapsulating them in the pay load of packet explicitly address to the foreign agent, instead of just sending them to the mobile users home.

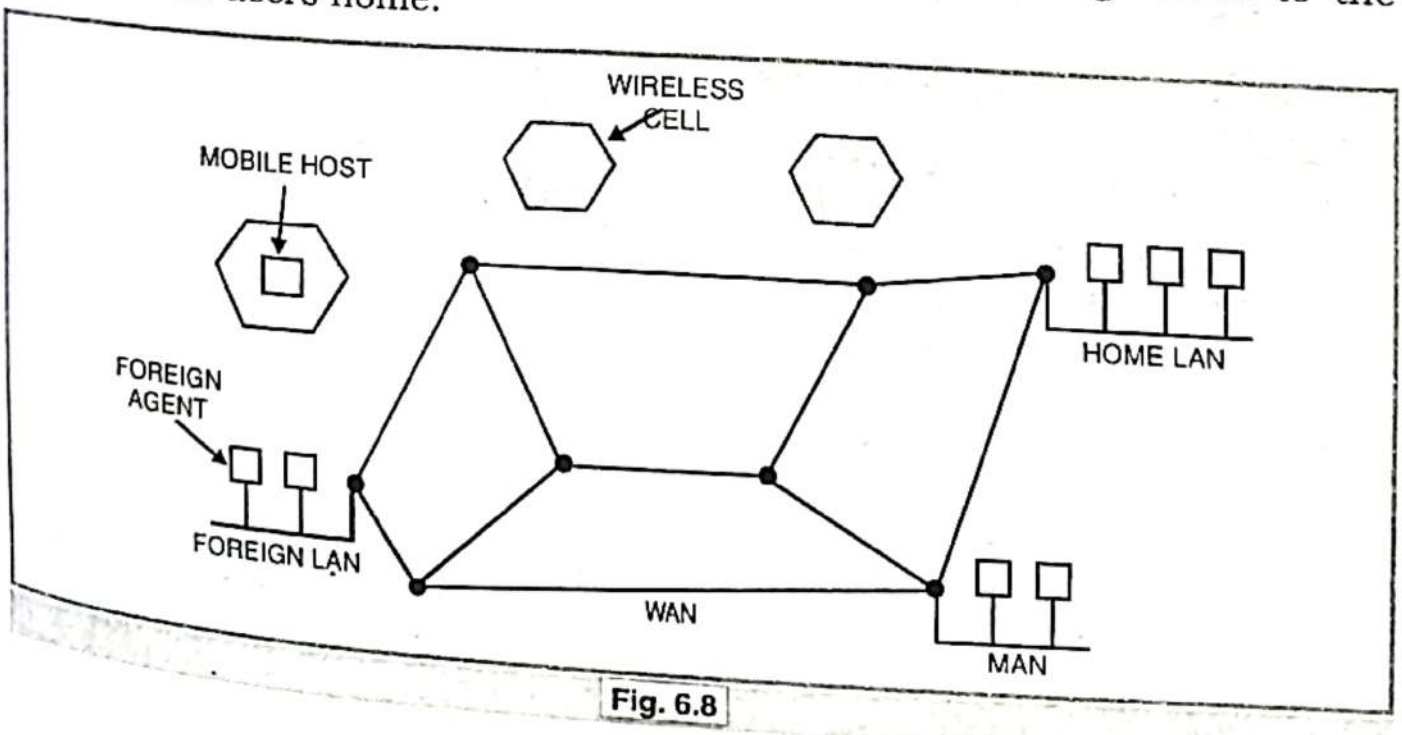


Fig. 6.8

The various schemes that have been proposed differ in several ways.

1. There is issue that how much of this protocol is carried out by the routers and how much by the hosts.
2. Routers along the way record mapped address so they can intercept and redirect traffic even before it get to the home location.
3. In some scheme each visitor is given a unique temporary address, they temporary address refers to an agent that handles traffic for all visitor.
4. The schemes differ in how they actually manage to arrange for packets that are addressed to one destination to be delivered to a different one.

6.8.12 Broadcast Routing

- Host need to send messages to many or all other hosts.
- Sending a packet to all destination simultaneously is called broadcasting. Various methods have been proposed for doing so.

1. Broadcasting method require no special feature from the subnet is for the source to simply sent a distinct packet to each destination. Not only is the wasteful of band width, but it also require the source to have a complete list of all destination.
2. Flooding is another method which is ill-suited for ordinary point to point communication for broadcasting it might rate serious consideration.. The problem with flooding as broadcast technique is the same problem.

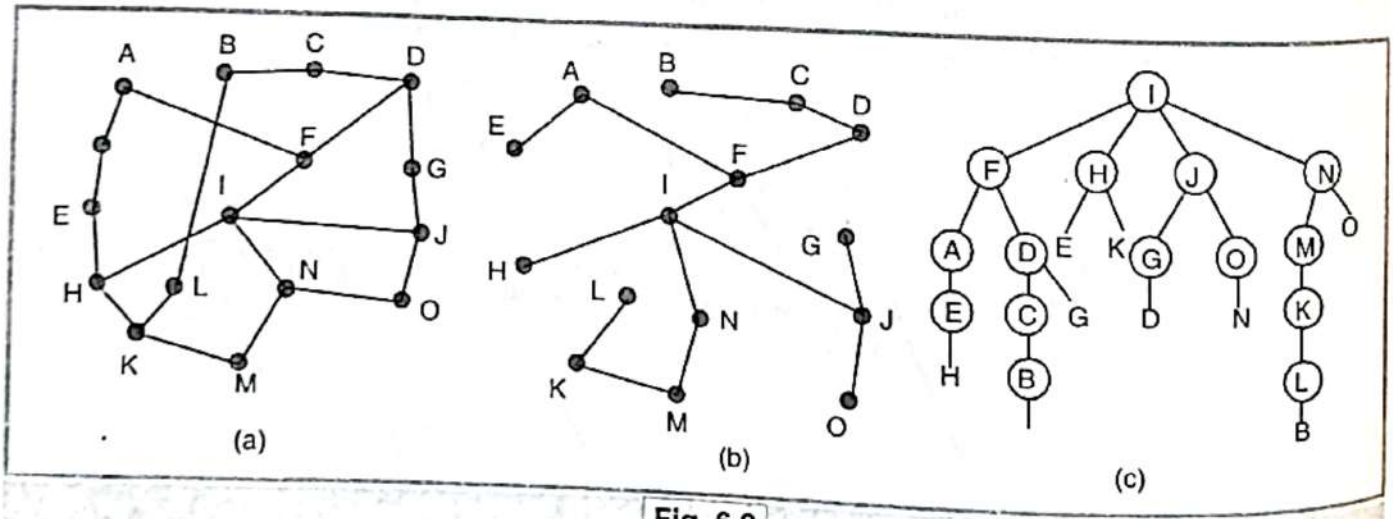


Fig. 6.9

3. In multidestination routing each packet contains either a list of destinations. When a packet arrives at a router, the router checks all destination to determine the set of output lines that will be needed. The router generates a new copy of the packet for each output line to

be used. Multidestination routing is like separately addressed packets, except that when several packets must follow the same route.

4. A broadcast algorithm make explicit use of the sink tree for the router initiating the broadcast. A spanning tree is a subnet of the subnet that includes all the routers but contains no loop. In each router knows which of its lines belong to the spanning tree, it can copy an incoming broadcast packet on to all the spanning tree lines except the one it arrived on.

In reverse path forwarding, on the first hop, I sends packet F,H J and N. On the second hop eight packets are generated two by each router that received a packet on first hop.

6.8.13. Multicast Routing

- For some applications, widely separated processes work together in groups.
- Sending a message to such group is called multicasting and its routing algorithm is called multicast routing.
- To do multicasting group management is required.
- Some way is needed to create and destroy groups and for process to join and leave group.
- To do multicast routing each router computers a spanning tree covering all other router in the subnet.
- When a process sends a multicast packet to a group, the first router examines its spanning tree and remove all lines that do not lead to hosts.
- Various ways for accessing the spanning tree are possible.
- The simplest one can be used if link state routing is used, each router is aware of complete subnet topology include the hosts of a particular group. One potential disadvantage of this algorithm is that it scales poorly to large networks.

6.8.14 Shortest Path Routing (Least-Cost-Path Algorithms)

- In practice, the majority of Internet routing methods are based on least-cost algorithms. In such algorithms, a link cost is proportional to the link's current traffic load.
- However, the link cost may not always be proportional to the current load.

- The link cost is defined on both directions between each pair of nodes.
- Several least-cost path algorithms have been developed for packet-switched networks.
- In particular, **Dijkstra's algorithm and the Bellman-Ford algorithm** are the most effective and widely used algorithms.
- Each arc of the graph representing a communication line (often called a link) to choose a route between a given pair of routers, the algorithm just finds the shortest path (least cost path) between them on the graph.
- In the most general case, the labels on the arcs could be computed as a function of the distance, bandwidth, average traffic, communication cost, mean queue length, measured delay, and other factors.
- By changing the weighting function, the algorithm would then compute the "shortest" path measured according to any one of a number of criteria, or a combination of criteria.

6.8.14.1 Dijkstra's Algorithm

- Dijkstra's algorithm is a centralized routing algorithm that maintains information in a central location.
- The objective is to find the least-cost path from a given source node to all other nodes.
- This algorithm determines least-cost paths from a source node to a destination node by optimizing the cost in multiple iterations. Dijkstra's algorithm is as follows :

Begin Dijkstra's Algorithm

1. Define : s = Source node
   k = Set of visited nodes by the algorithm
   α_{ij} = Cost of the link from node i to node j
   β_{ij} = Cost of the least-cost path from node i to node j
2. Initialize : $k = \{s\}$
   $\beta_{sj} = \alpha_{sj}$ for $j \neq s$
3. Next node: Find x k that $\beta_{sx} = \min \beta_{sj}$ for j k.
   Add x to k.
4. Least-cost paths : $\beta_{sj} = \min (\beta_{sj}, \beta_{sx} + \alpha_{xj})$ for j k

- If any two nodes i and j are not connected directly, the cost for that link is infinity, indicated by $\beta_{ij} = x$.
- Steps 2 and 3 repeated until paths are assigned to all nodes.
- At step 1, k represents s, and β_{sj} computes the cost of the least-cost path from s to node j.
- At step 2, we want to find x among the neighbouring nodes but not in k such that cost is minimized.
- At step 3, we simply update the least-cost path.
- The algorithm ends when all nodes have been visited and included in the algorithm.

Example 1. Using Dijksra's algorithm, find the least-cost path from node A to node B in Fig. 8

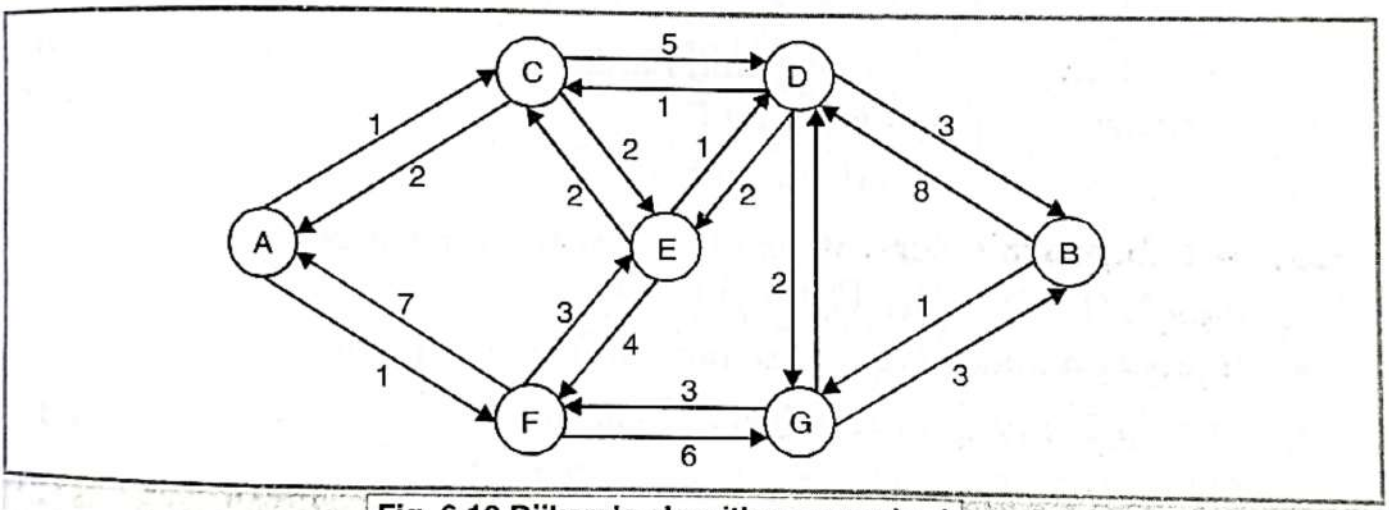


Fig. 6.10 Dijksra's algorithm example. 1

Solution

The detailed operation is shown in the Table. 6.2. The first step is to find a path from the source node A to all other nodes. Thus, at the first row, k = {A}. It is obvious there are direct links from A to nodes C and F. Therefore, the least-cost path for either node is 1, as shown in fig. 8. We then fill the table with AC(1) and AF(1), respectively. Given k = {A}, there is no connections between A and nodes D, E, G, and B. The algorithm continues until all nodes have been included as k = A,C,F,E, D,G,B, and we obtain the least-cost path of ACEDB(7).

Table 6.1 Use of Dijkstra's algorithm

K	β_{AC}	β_{AF}	β_{AE}	β_{AD}	β_{AG}	β_{AB}
{A}	AC(1)	AF(1)	x	x	x	x
{A,C}	AC(1)	AF(1)	ACE(3)	ACD(6)	x	x

{A,C,F}	AC(1)	AF(1)	ACE(3)	ACD(6)	AFG(7)	x
{A,C,F,E}	AC(1)	AF(1)	ACE(3)	ACED(4)	AFG(7)	x
{A,C,F,E,D}	AC(1)	AF(1)	ACE(3)	ACED(4)	ACEDG(6)	ACEDB(7)
{A,C,F,E,D,G}	AC(1)	AF(1)	ACE(3)	ACED(4)	ACEDG(6)	ACEDB(7)
{A,C,F,E,D,G,B}	AC(1)	AF(1)	ACE(3)	ACED(4)	ACEDG(6)	ACEDB(7)

6.8.14.2 Bellman-Ford Algorithm

- The Bellman-Ford algorithm finds the least-cost path from a source to a destination by passing through no more than/links.
- The essence of the algorithm consists of the following steps :

Begin Bellman-Ford Algorithm

1. Define : s = Source node

   α_{ij} = Cost of the link from node i to node j

   $\beta_{ij}(l)$ = Cost of the least-cost path from i to j with no more than l links

2. Initialize : $\beta_{sj}(0) = x$, for all j s

   $\beta_{ss}(0) = 0$, for all l

3. Least-cost paths : for any node j $\neq$ s with predecessor node i:

   $\beta_{sj}(l+1) = \min_i [\beta_{si}, (l) + \alpha_{ij})]$

- If any two nodes i and j are not connected directly, $\beta_{ij}(l) = x$.
- At step 2, every value of β is initialized.
- At step 3, we increase the number of links l in a sequence of iterations.
- During each iteration, we find the least-cost path, given the value of l.
- The algorithm ends when all nodes have been visited and included in the algorithm.

Example 2. Use Bellman-Ford algorithm to find the least-cost path from node A to node B in Fig. 9.

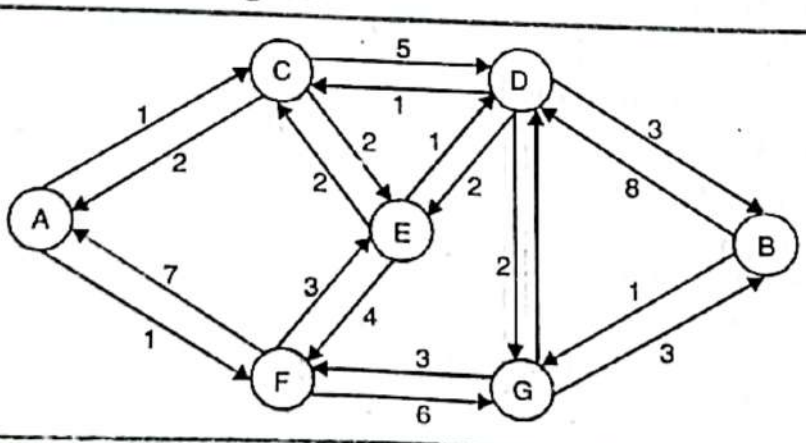


Fig. 6.11 Dijksra's algorithm example. 2

Table 2. Shows the details of least-cost-path iterations. For iteration l = 1, only AC with cost 1 and AF with cost 1 exist, owing to the restriction enforced by l = 1. This trend changes at iteration l = 2, when ACE with cost 3 can be added to he set of least-cost paths. As seen, the result of the final least-cost path is identical to the one obtained by Dijkstra's algorithm.

Table 6.2 Use of the Bellman-Ford algorithm

l	β_{AC}	β_{AF}	β_{AE}	β_{AD}	β_{AG}	β_{Ab}
0	x	x	x	x	x	x
1	AC(1)	AF(1)	x	x	x	x
2	AC(1)	AF(1)	ACE(3)	ACD(6)	AFG(7)	x
3	AC(1)	AF(1)	ACE(3)	AFD(4)	AFG(7)	ACDB(9)
4	AC(1)	AF(1)	ACE(3)	AED(4)	ACEG(6)	ACEDB(7)

We can now compare these two algorithms. In step 2 of Bellman-Ford, the calculation of the link cost to any node j requires knowledge of the link cost to all neighboring nodes. In step 3 of Dijkstra's algorithm, each node requires knowledge of the network topology at each time of iteration. The performance of each algorithm varies network to network and depends on the topology and size of a particular network. The comparison of these two algorithms, therefore, depends on the speed of each to achieve its objective at its corresponding step given a network under routing.

6.9 CONGESTION

- The performance of a subnet gets degraded in term soft speed and reliability both.
- If too many packets are present in the subnet.
- This situation of subnet is known as congestion or it is said that subnet is gone congested.
- The situation arises when number of packets becomes so large that it is out of carrying capacity of the subnet.
- In such cases the nodes are no longer able to cope and they begin losing packets.
- If the traffic is very high, some times the entire system collapses completely with no delivery of packets at all.

Various factors causing congestion are as follows :

* Slow working of nodes in updating tables and queuing buffers
* Input traffic rate is greater than output lines.

* Low buffer capacity of nodes
* Low adsorbing capacity of subnet as compared to pumping capacity of host.

- Congestion always has a tendency to feed itself causing a worse conduction of sub networking.
- If a node is slow then it sends a packet slowly as all the jobs performed by it are slow.
- It can not discard that packet unless it gets an acknowledgement from receiver.
- It has now a lesser buffer space for incoming packets.
- If a packet is discarded by receiver it retransmits it and may be many times causing more delay.
- If there is already a congestion at receiver it also works slow and the senders buffer is busy with a packet unnecessary.
- This is how congestion backs up.
- When congestion increases very much the movements of packets across subnet can stop altogether causing a deadlock.
- It is a condition that must never occur.

There are several ways to avoid dead lock. They are as follows:

* By monitoring vacant buffer at each node so that inward flow of packets can be stopped when buffer nears completely full.
* By discarding some packets and for this a life time of packet should be fixed. It does not happen that if a packet is again and again discarded by receiver it is again and again retransmitted by sender. There should be a life time for a packet and then it should be automatically discarded freeing the buffer.
* By dividing the buffers at each node into several classes, so that if a receiver is not able to accept a packet at same level because of unavailability of buffer the next higher level buffer can be used for that.
* By pre allocation of buffers i.e. on each call request the packet reserves one or more data buffers for it at destination. In such case if a call request arrives at a node and all buffers at node are already full a busy signal is returned to caller. Even if buffers are empty and are less than the required number a busy signal is returned to caller. This buffer reserving does not add any new problems to the situation that were not already there.

* By limiting the number of packets in the subnet, the congestion can be controlled. The method is known as isoarithmic as it keeps number of packets, in a subnet, constant. For that each sender, before transmitting a packet, has to get a permit for that. The source before sending a packet captures a permit and destroys it. On the other hand the receiver after removing the packet from subnet, regenerates the permit. This ensures the condition of deadlock will never occur in the subnet.
* Some times flow control is also used to prevent from dead lock. Though flow control cannot really solve the problem of congestion but as it prevents the node from saturating its neighbors so a stride-flow control rule can prevent the subnet from heavy loading.

- By using one or more methods described above the congestion and deadlocks can be controlled in a subnet.
- Deadlocks are ultimate congestion and are also known as lock ups.
- The simplest lock up happens with two nodes in which both want to send packets to each other but as the buffers of them are full no one can proceed i.e. both are stuck.
- Similarly if more than one nodes are involved in the same manner then also there is a dead lock.
- The previous one is known as direct store-and-forward lock up and the later one is known as indirect store-and-forward lock up.
- Actually the host is waiting for an acknowledgement and is not able to Free its buffer as the packet are not arrived safely at their destination.
  - Congestion is an important issue that can arise in packet switched network.
  - Congestion is a situation in computer networks in which the performance of network is degraded due to the presence of too many packets in the subnet.
  - Congestion in a network may occur when the load on the network (*i.e.* the number of packets sent to the network) is greater than the capacity of the network (*i.e.* the number of packets a network can handle.)
  - Figure 6.12 shows the concept of congestion in a subnet.
  - As shown in fig. 6.12 when the number of packets dumped into the subnet by the hosts is within its carrying capacity, they are all

delivered. At this stage number of packets delivered is proportional to the number of packets sent and no congestion take place.

- As the traffic increases too far, the routers are no longer able to cope up and they start losing packets with further increase in the traffic, performance degrades more and more, packets are lost and congestion worsens.
- At very high traffic, performance collapses completely and no packets are delivered (see fig. 6.12)

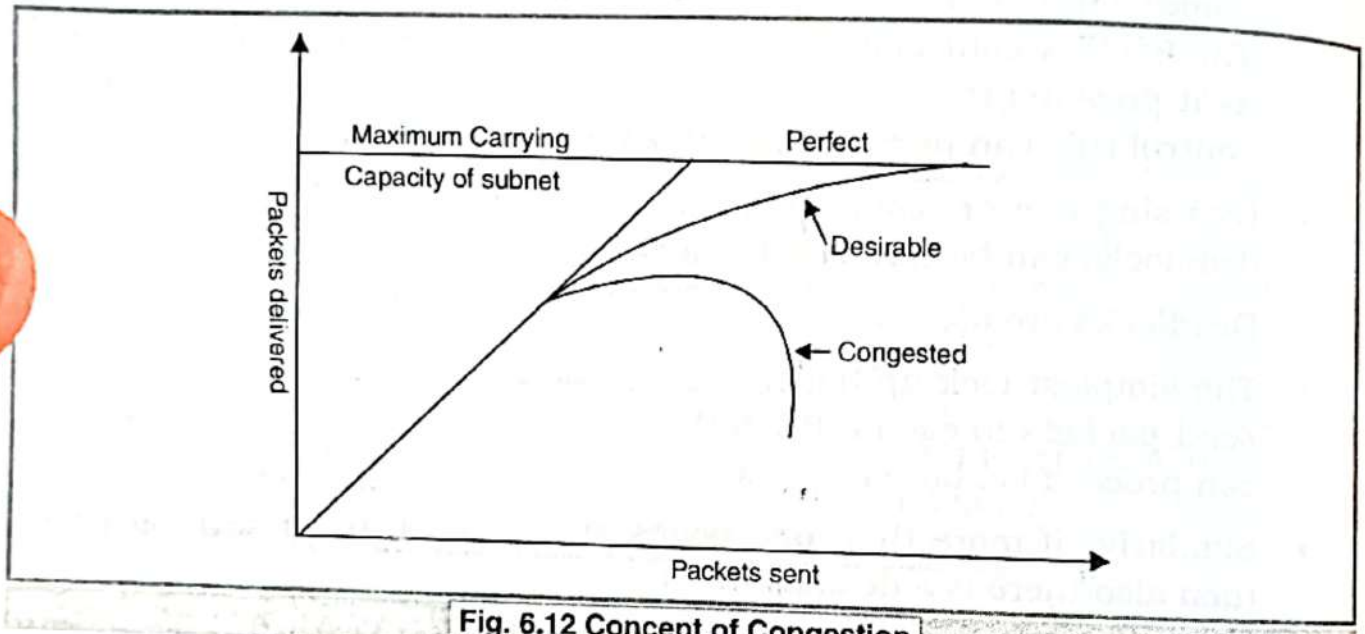


Fig. 6.12 Concept of Congestion

6.9.1 Causes of Congestion

The various causes of congestion in a subnet are :

1. If suddenly, a stream of **packet start arriving on three or four input lines** and all need the same output line (see fig. 6.13). In this case, a queue will be built up.

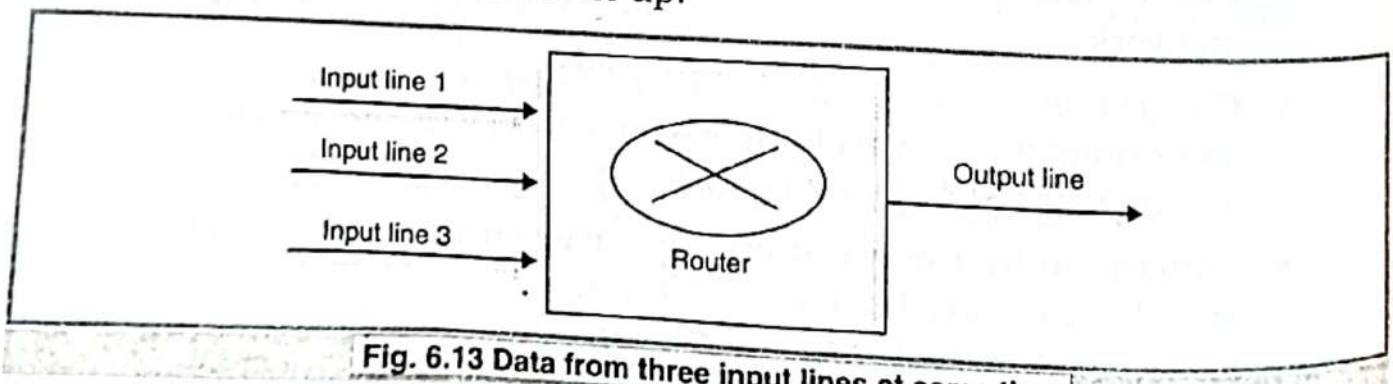


Fig. 6.13 Data from three input lines at same time

If there is insufficient memory to hold all the packets, the packet will be lost.

Increasing the memory to unlimited size does not solve the problem. This is because, by the time packets reach front of the queue, they

have already timed out (as they waited in the queue). When timer goes off source transmits duplicate packet that are also added to the queue. Thus same packets are added again and again, increasing the load all the way to the destination.

2. Congestion in a subnet can occur if the **processors are slow.** Slow speed CPU at routers will perform the routine tasks such as queuing buffers, updating table etc slowly. As a result of this, queues are built up even though there is excess line capacity.
3. Congestion is also caused by **slow links.** This problem will be solved when high speed links are used. But it is not always the case. Sometimes increase in link bandwidth can further deteriorate the congestion problem as higher speed links may make the network more unbalanced.
4. **Congestion can make itself worse.** If a router does not have free buffers, it start ignoring/discarding the newly arriving packets. When these packets are discarded, the sender may retransmit them after the timer goes off. Such packets are transmitted by the sender again and again until the source gets the acknowledgement of these packets. Therefore multiple transmissions of packets will force the congestion to take place at the sending end.

6.9.2 Congestion Control

- **Congestion Control** refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened.
- Congestion control mechanisms are divided into two categories, one category prevents the congestion from happening and the other category removes congestion after it has taken place.

These two categories are (see fig. 6.14):

1. Open loop
2. Closed loop

Open Loop Congestion Control

- In this method, policies are used to prevent the congestion before it happens.
- Congestion control is handled either by the source or by the destination.

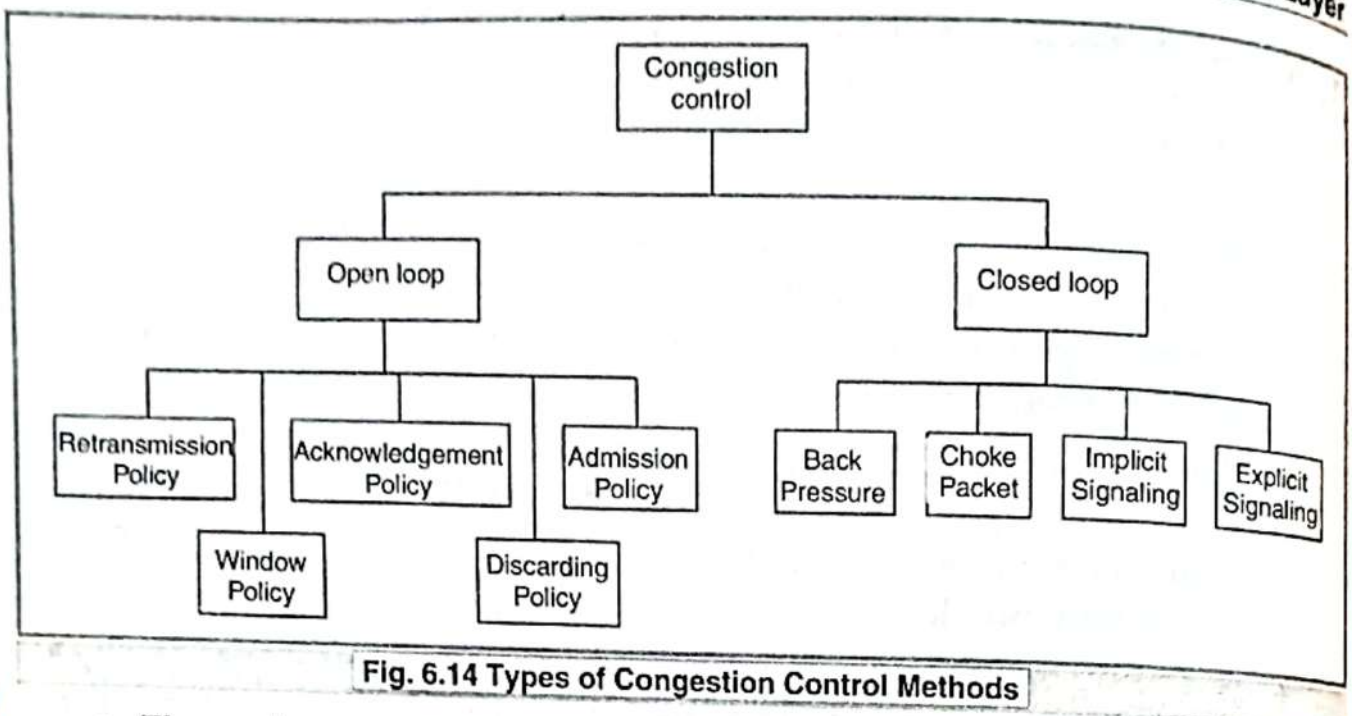


Fig. 6.14 Types of Congestion Control Methods

- The various methods used for open loop congestion control are :

1. Retransmission Policy

We can avoid this and prevent congestion

- The sender retransmits a packet, if it feels that the packet it has sent is lost or corrupted.
- However retransmission in general may increase the congestion in the network. But we need to implement good retransmission policy to prevent congestion.
- The retransmission policy and the retransmission timers need to be designed to optimize efficiency and at the same time prevent the congestion.
- The type of windows at the sender may also effect congestion. The relative report window is better than the go back to window.

2. Window Policy

- To implement window policy, selective reject window method is used for congestion control.
- Selective Reject method is preferred over Go-back-n window as in Go-back-n method, when timer for a packet times out, several packets are resent, although some may have arrived safely at the receiver. Thus, this duplication may make congestion worse.
- Selective reject method sends only the specific lost or damaged packets.

3. Acknowledgement Policy

- The acknowledgement policy imposed by the receiver may also affect congestion.
- If the receiver does not acknowledge every packet it receives it may slow down the sender and help prevent congestion.
- Acknowledgments also add to the traffic load on the network. Thus, by sending fewer acknowledgements we can reduce load on the network.
- To implement it, several approaches can be used :
  1. A receiver may send an acknowledgement only if it has a packet to be sent.
  2. A receiver may send an acknowledgement when a timer expires.
  3. A receiver may also decide to acknowledge only *N* packets at a time. High flow control to avoid congestion.

4. Discarding Policy

- A router may discard less sensitive packets when a congestion is likely to happen.
- Such a discarding policy may prevent congestion and at the same time may not harm the integrity of the transmission.

5. Admission Policy

- An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual circuit networks.
- Switches in a flow first check the resource requirement of a flow before admitting it to the network.
- A router can deny establishing a virtual circuit connection if there is congestion in the network or if there is a possibility of future congestion.

Closed Loop Congestion Control

- Closed loop congestion control mechanisms try to remove the congestion after it happens.
- The various methods used for closed loop congestion control are :

1. Backpressure

- Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source as shown in figure 6.15

- The backpressure technique can be applied only to virtual circuit networks. In such virtual circuit each node knows the upstream node from which a data flow is coming.

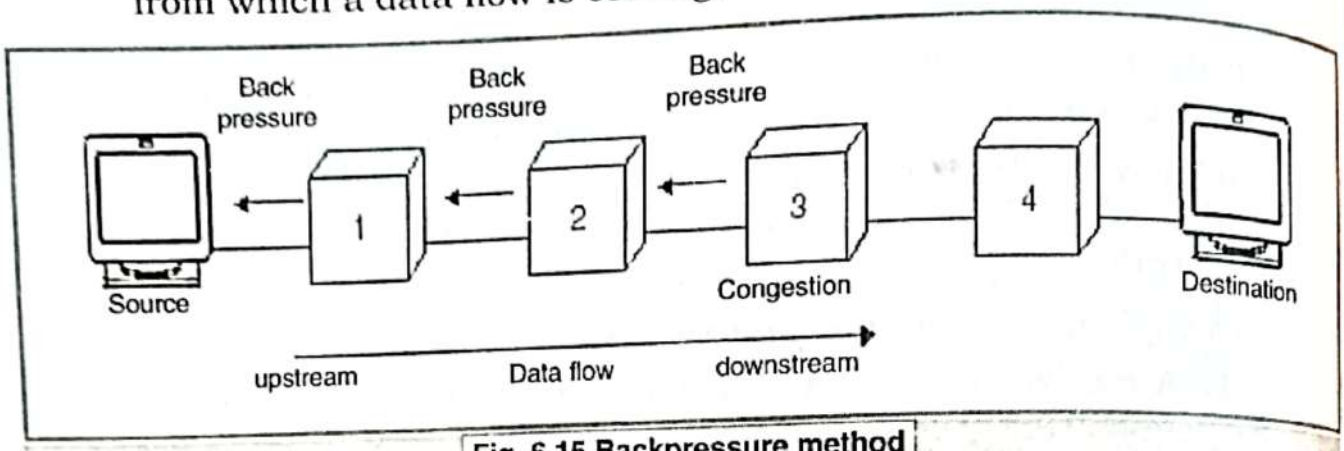


Fig. 6.15 Backpressure method

- In this method of congestion control, the congested node stops receiving data from the immediate upstream node or nodes.
- This may cause the upstream node on nodes to become congested, and they, in turn, reject data from their upstream node or nodes.
- As shown in fig. 6.21 node 3 is congested and it stops receiving packets and informs its upstream node 2 to slow down. Node 2 in turns, may be congested and informs node 1 to slow down. Now node 1 may create a congestion and informs the source node to slow down. In this way the congestion is alleviated. Thus, the pressure on node 3 is moved backward to the source to remove the congestion.

2. Choke Packet

- In this method of congestion control, congested router or node sends a special type of packet called **choke packet** to the source to inform it about the congestion.
- Here, congested node does not inform its upstream node about the congestion as in backpressure method.

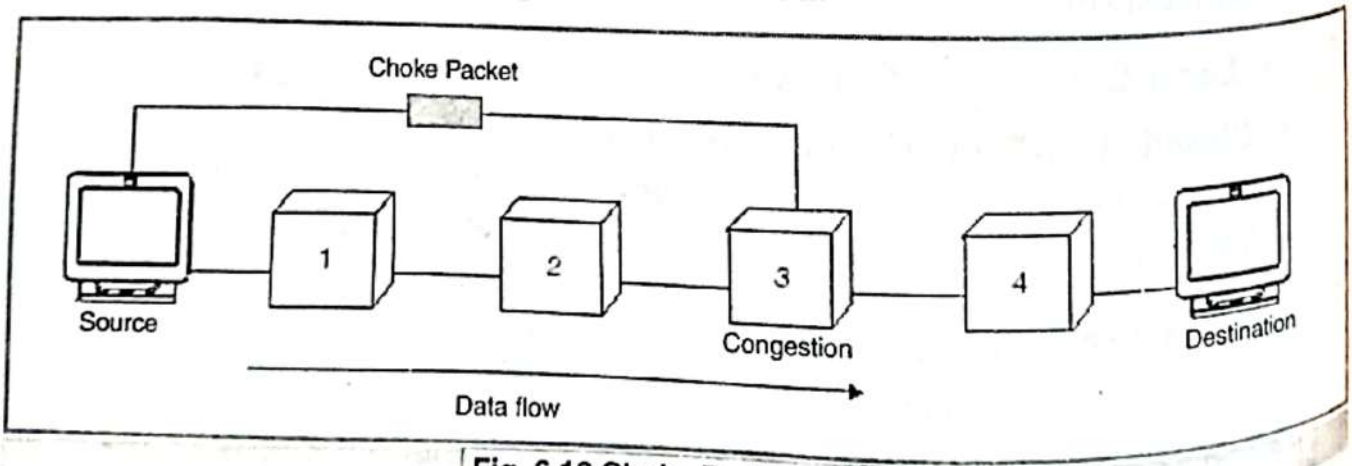


Fig. 6.16 Choke Packet Method

- In choke packet method, congested node sends a warning directly to the source station *i.e.* the intermediate nodes through which the packet has traveled are not warned (see fig. 6.16)

3. Implicit Signaling

- In implicit signaling, there is no communication between the congested node or nodes and the source.
- The source guesses that there is a congestion somewhere in the network when it does not receive any acknowledgment. Therefore the delay in receiving an acknowledgment is interpreted as congestion in the network.
- On sensing this congestion, the source slows down.
- This type of congestion control policy is used by TCP.

4. Explicit Signaling

- In this method, the congested node explicitly send a signal to the source or destination to inform about the congestion.
- Explicit signaling is different from the choke packet method. In choke packed method, a separate packet is used for this purpose whereas in explicit signaling method, the signal is included in the packets that carry data.
- Explicit signaling can occur in either the forward direction or the backward direction.
- In **backward signaling,** a bit is set in a packet moving in the direction opposite to the congestion. This bit warns the source about the congestion and informs the source to slow down.
- In **forward signaling**, a bit is set in a packet moving in the direction of congestion. This bit warns the destination about the congestion. The receiver in this case uses policies such as slowing down the acknowledgements to remove the congestion.

6.9.3 Congestion control policies used in different layers

Different layers of OSI model use different policies or techniques to control the congestion in subnet. The various policies used by data link layer, network layer and transport layer are shown in the table below :

Layer	Policies
Data Link Layer	1. Retransmission policy
	2. Out of order policy

	3. Acknowledgement policy
	4. Flow control policy
Network layer	1. Virtual circuit Vs Datagrams inside the subnet
	2. Packet queuing and service policy
	3. Packet discard policy
	4. Routing algorithm
	5. Packet lifetime management
Transport Layer	1. Transmission policy
	2. Out-of-order caching policy
	3. Acknowledgement policy
	4. Flow control policy
	5. Time–out determination

6.9.4 Congestion control algorithms

1. Leaky Bucket Algorithm

- It is a traffic shaping mechanism that controls the amount and the rate of the traffic sent to the network.
- A leaky bucket algorithm shapes bursty traffic into fixed rate traffic by averaging the data rate.
- Imagine a bucket with a small hole at the bottom as shown in fig. 6.17.

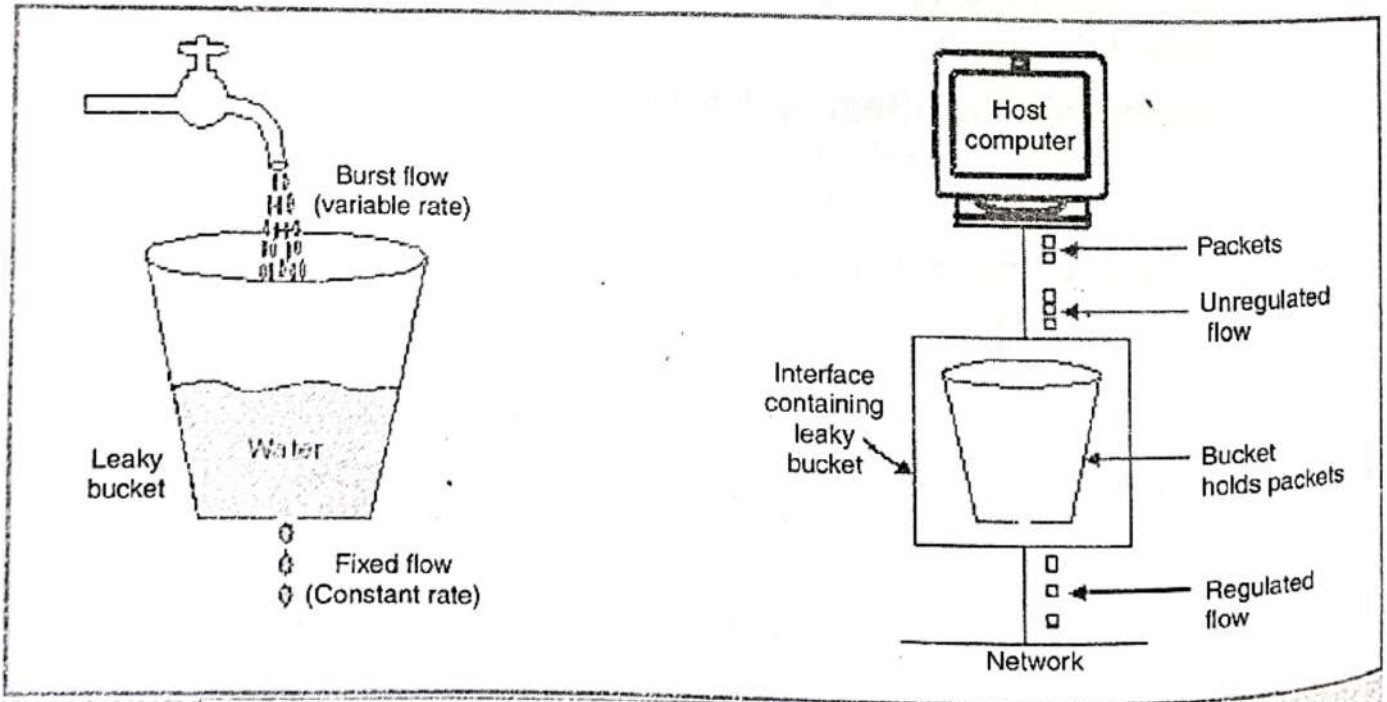


Fig. 6.17 (a) A leaky bucket with water (b) A leaky bucket with packets

- The rate at which the water is poured into the bucket is not fixed and can vary but it leaks from the bucket at a constant rate. Thus (as

long as water is present in bucket), the rate at which the water leaks does not depend on the rate at which the water is input to the bucket.

- Also, when the bucket is full, any additional water that enters into the bucket spills over the sides and is lost.
- The same concept can be applied to packets in the network (See fig. 6.17 (b))

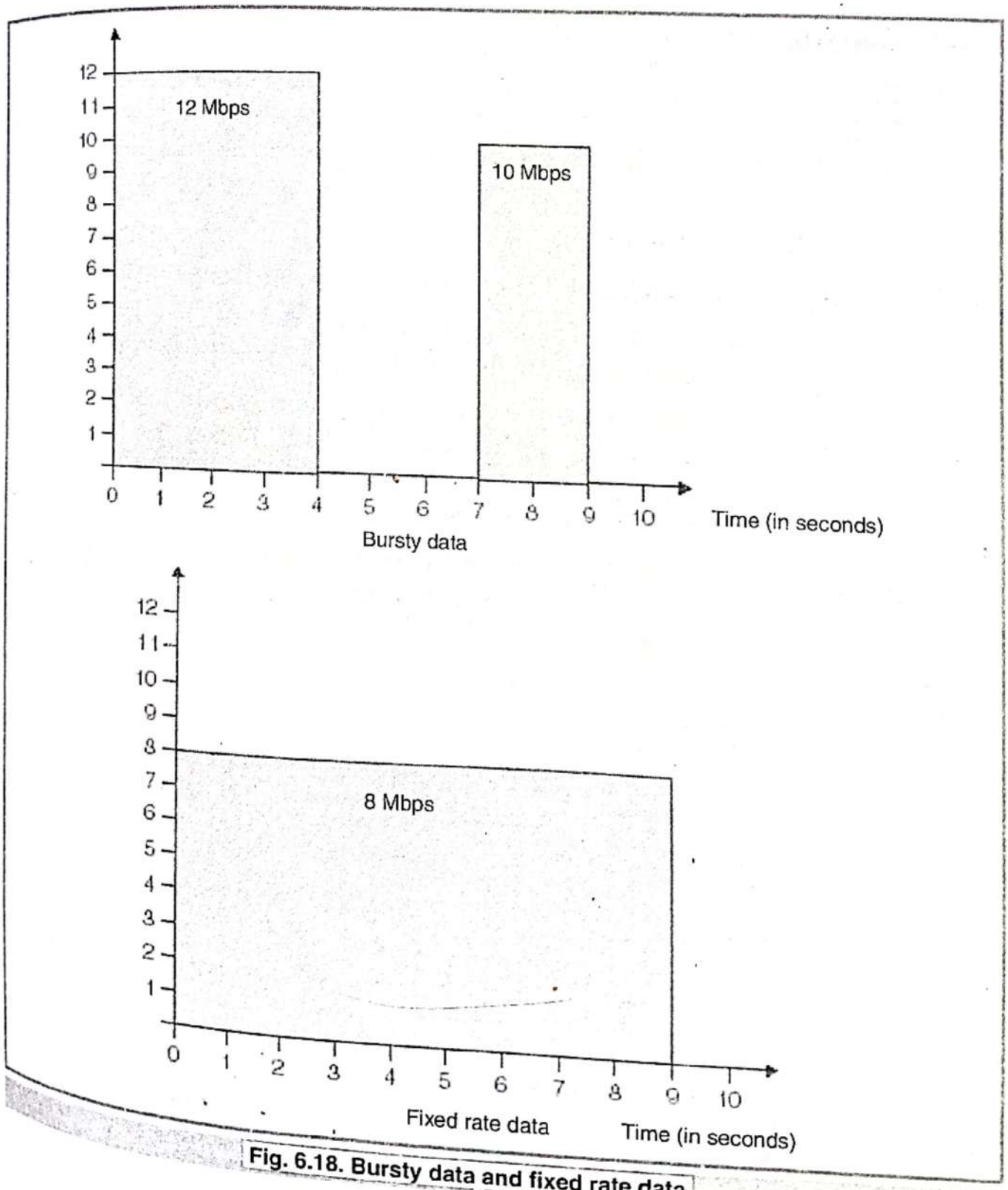


Fig. 6.18. Bursty data and fixed rate data

Consider that data is coming from the source at variable speeds. Suppose that a source sends data at 12 Mbps for 4 seconds. Then there is no data for 3 seconds. The source again transmits data at a rate of 10 Mbps for 2 seconds. Thus, in a time span of 9 seconds, 68 Mb data has been transmitted.

If a leaky bucket algorithm is used, the data flow will be 8 Mbps for 9 seconds. Thus constant flow is maintained (see fig. 6.18)

Implementation of Leaky Bucket Algorithm

1. Each host is connected to the network by an interface containing a leaky bucket. A leaky bucket is implemented by using a first in first out (FIFO) queue that holds the packet.
2. If a packet arrives at the queue when it is full, the packet is discarded. (see fig. 6.19)
3. It make use of clock tick to remove the packets from FIFO queue.
4. If the traffic consists of fixed size packets, the process removes fixed number of packets from the queue at each tick of the clock.
5. If packet sizes are not same, then the system allows fixed number of bytes to be removed at each tick of the clock. For example if the rule is 1024 bytes per tick, a single 1024-byte packet, two 512-byte packets or four 256-byte packets can be removed per tick.
6. Thus, a leaky bucket algorithm turns an uneven flow of packets from the user processes inside the host into an even flow of packets onto the network.
7. Such a system smooth out the bursty traffic and reduces the chances of congestion.

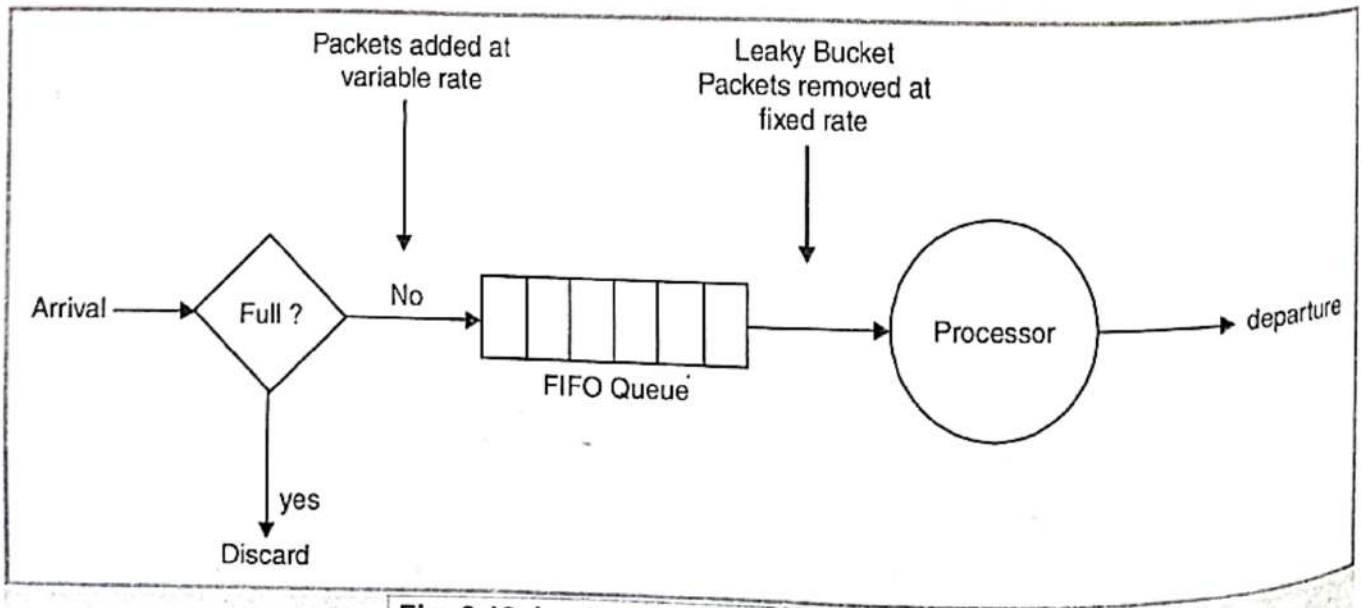


Fig. 6.19. Implementation of Leaky bucket

- If packets are of variable sizes then the following procedure is used by leaky bucket algorithm (See fig. 6.20)
  (i) A counter(n) is initialized to the amount of data that can be sent in one tick.
  (ii) The algorithm checks the size of packet at the front of the queue.
  (iii) If the size is less than or equal to the value of counter, the packet is sent and the counter (n) is decremented by the packet size. This step is repeated until n is smaller than packet size.

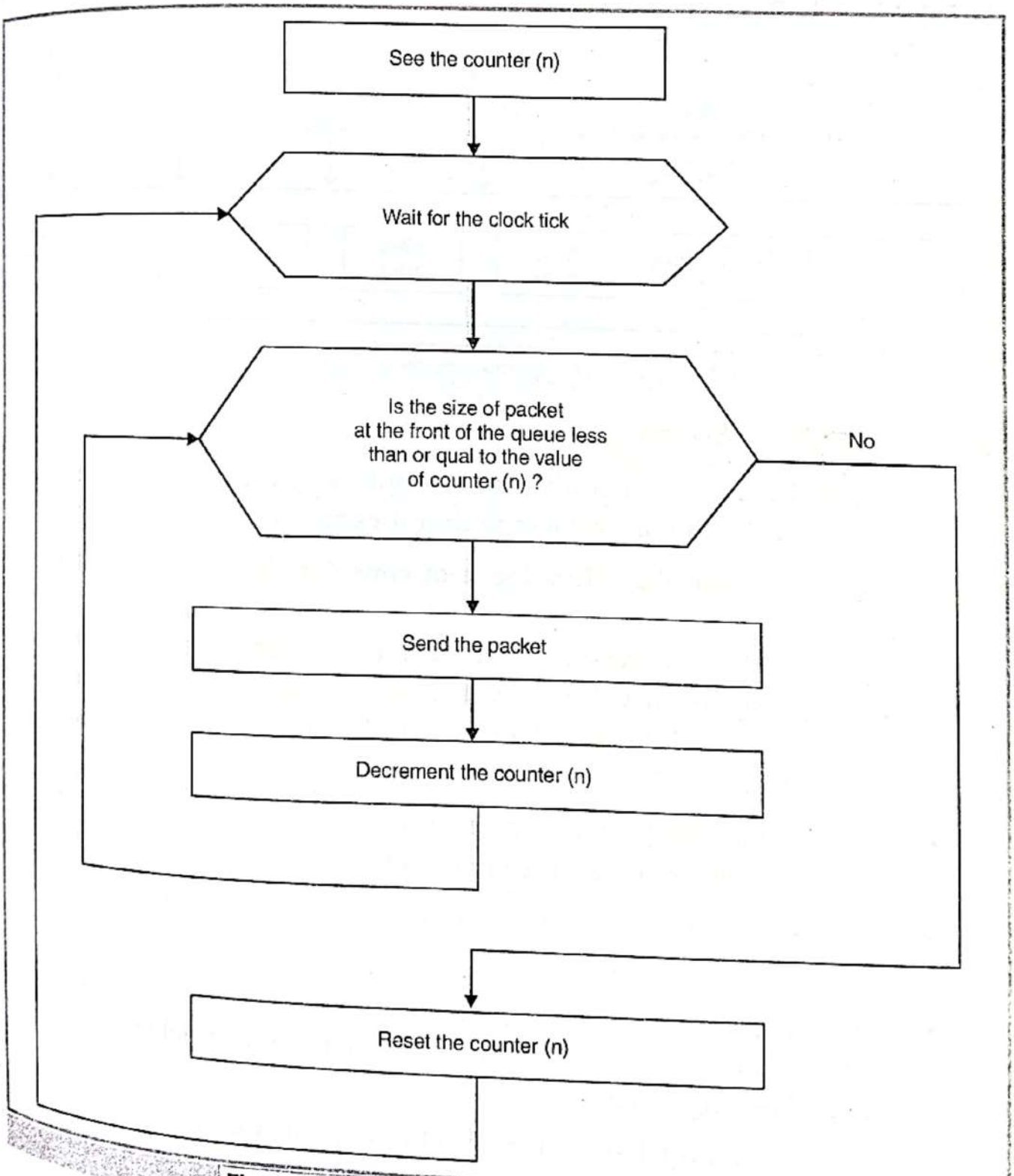


Fig. 6.20. Flowchart for leaky bucket algorithm

(iv) When the size of packet is greater than the value of counter (n), the packet is left in the queue and waits for the next tick of the clock.

Fig. 6.20 shows an example. Assume that the output rate is 80 kbps. This means 80,000 bits per second or 10,000 bytes per second.

The counter is initially set to 10,000; after sending three packets, the value of the counter is 600, which is less than the size of the next packet. The next three packets cannot be sent. They need to wait for the next tick of the clock.

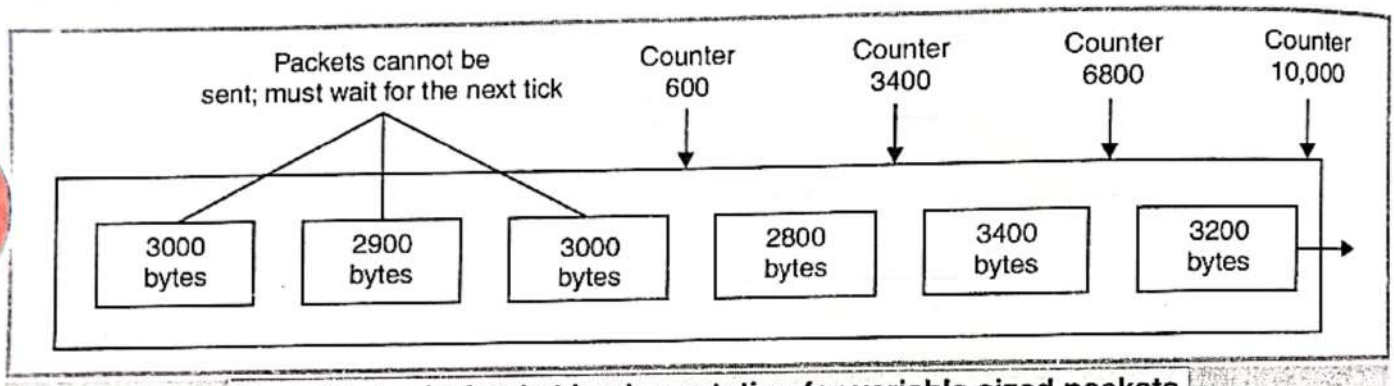


Fig. 6.21. Leaky bucket implementation for variable sized packets

2. Token bucket Algorithm

- The leaky bucket algorithm allows only an average (constant) rate of data flow. Its major problem is that it cannot deal with bursty data.
- A leaky bucket algorithm does not consider the idle time of the host. For example, if the host was idle for 10 seconds and now it is willing to sent data at a very high speed for another 10 seconds, the total data transmission will be divided into 20 seconds and average data rate will be maintained. The host is having no advantage of sitting idle for 10 seconds.
- To overcome this problem, a token bucket algorithm is used. A token bucket algorithm allows bursty data transfers.
- A token bucket algorithm is a modification of leaky bucket in which leaky bucket contains tokens.
- In this algorithm, a token(s) are generated at every clock tick. For a packet to be transmitted, system must remove token(s) from the bucket. (See fig. 6.22)
- Thus, a token bucket algorithm allows idle hosts to accumulate credit for the future in form of tokens.

- For example, if a system generates 100 tokens in one clock tick and the host is idle for 100 ticks. The bucket will contain 10,000 tokens.

Now, if the host wants to send bursty data, it can consume all 10,000 tokens at once for sending 10,000 cells or bytes.

Thus a host can send bursty data as long as bucket is not empty.

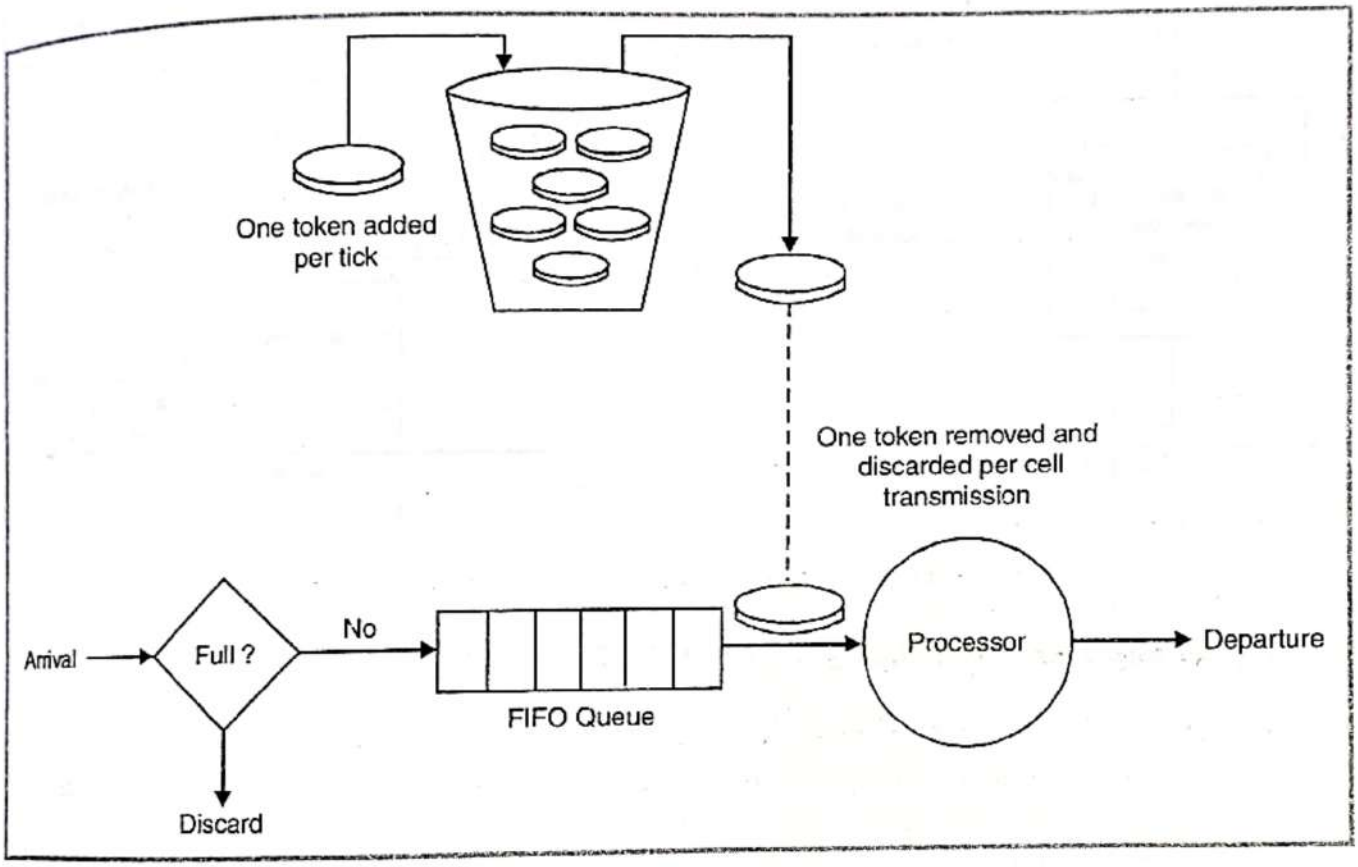


Fig. 6.22. Token bucket algorithm

Implementation of token bucket algorithm

1. This algorithm make use of a variable or counter that counts the token. This counter is initialized to zero.
2. The counter is incremented by 1, each time a token is generated.
3. Whenever a packet is sent, the counter is decremented by one.
4. When the counter becomes zero, no packets can be sent.

For example, as shown in fig. 6.23 (a), token bucket contains 5 tokens and 7 packets are waiting to be transmitted.

In order to get transmitted, each packet captures and destroy one token.

Fig. 6.23 (b) shows that 5 out of 7 packet have gotten through, but the other two are struck waiting for two more tokens to be generated.

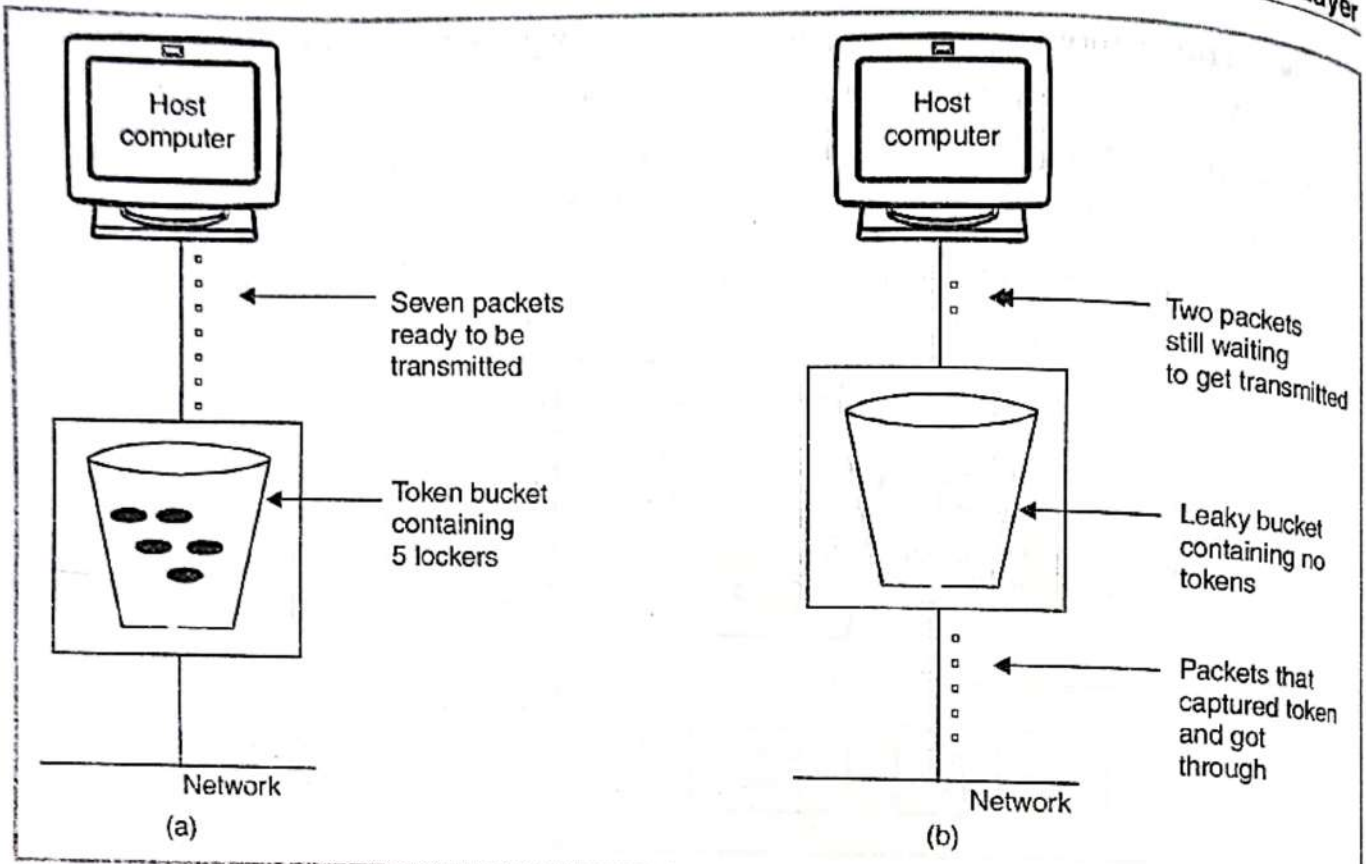


Fig. 6.23. Implementation of token bucket

Comparison between leaky bucket and token bucket algorithm

S.No.	Leaky Bucket	Token Bucket
1.	Leaky bucket is rigid algorithm as it outputs the data at an average rate and does not support bursty data.	Token bucket algorithm is flexible as it enables the bursty data to be sent immediately.
2.	It does not credit the idle time of the host *i.e.* it does not generate tokens.	It credits the idle time of the host and accumulates it in form of the tokens.
3.	The leaky bucket algorithm discards the incoming packets if the bucket (FIFO queue) is full.	The token bucket algorithm throws away tokens if bucket is full. It never discards packets when bucket is full.

6.10 CONGESTION CONTROL AT NETWORK LAYER

- Congestion represents an overloaded condition in a network.
- Congestion control can be achieved by optimum usage of the available resources in the network. Fig. 6.5 shows a set of performance graphs for networks in which three possible cases are compared : no congestion, moderate congestion, and severe congestion.

- These plots indicate that if a network has no congestion control, the consequence may be a severe performance degradation, even to the extent that the carried load starts to fall with increasing offered load.
- The ideal situation is the one with no loss of data, as a result of no congestion.
- Normally, a significant amount of engineering effort is required to design a network with no congestion.

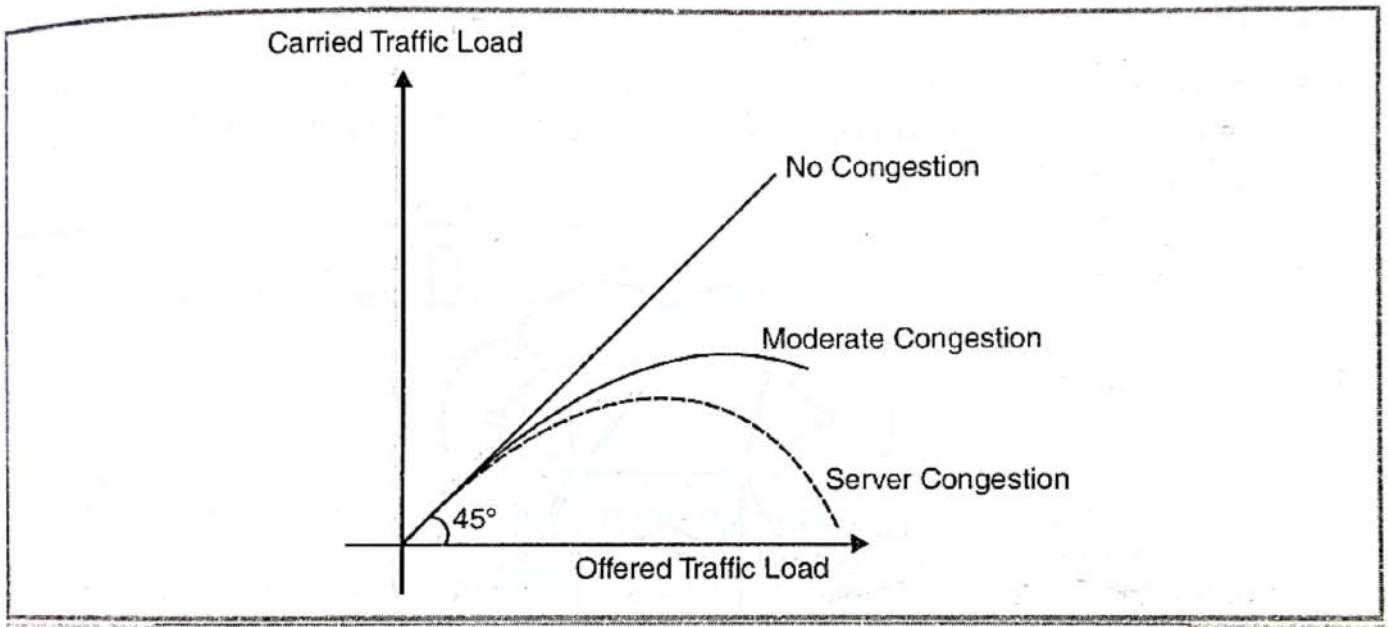


Fig. 6.24 Comparison among networks in which congestion, moderate congestion, and severe congestion exist

- Congestion can be either logical or physical, as shown in Fig. 6.25.
- The queueing feature in two routers can create a logical bottleneck between user A and user B.
- Meanwhile, insufficient bandwidth a resource shortage on physical links between routers and the network can also be a bottleneck, resulting in congestion. Resource shortage can occur.
  - At the link layer, where the link bandwidth runs out
  - At the network layer, where the queues of packets at nodes go out of control
  - At the transport layer, where logical links between two routers within a communication session go out of control.
- One key way to avoid congestion is to carefully allocate network resources to users and applications.
- Network resources, such as bandwidth and buffer space, can be allocated to competing applications.

- Devising optimum and fair resource-allocation schemes can control congestion to some extent. In particular, congestion control applies to controlling data flow among a group of senders and receivers, whereas flow control is specifically related to the arrangement of traffic flows on links.
- Both resource allocation and congestion control are not limited to any single level of the protocol hierarchy.
- Resource allocation occurs at switches, routers, and end hosts.
- A router can send information on its available resources so that an end host can reserve resources at the router to use for various applications.

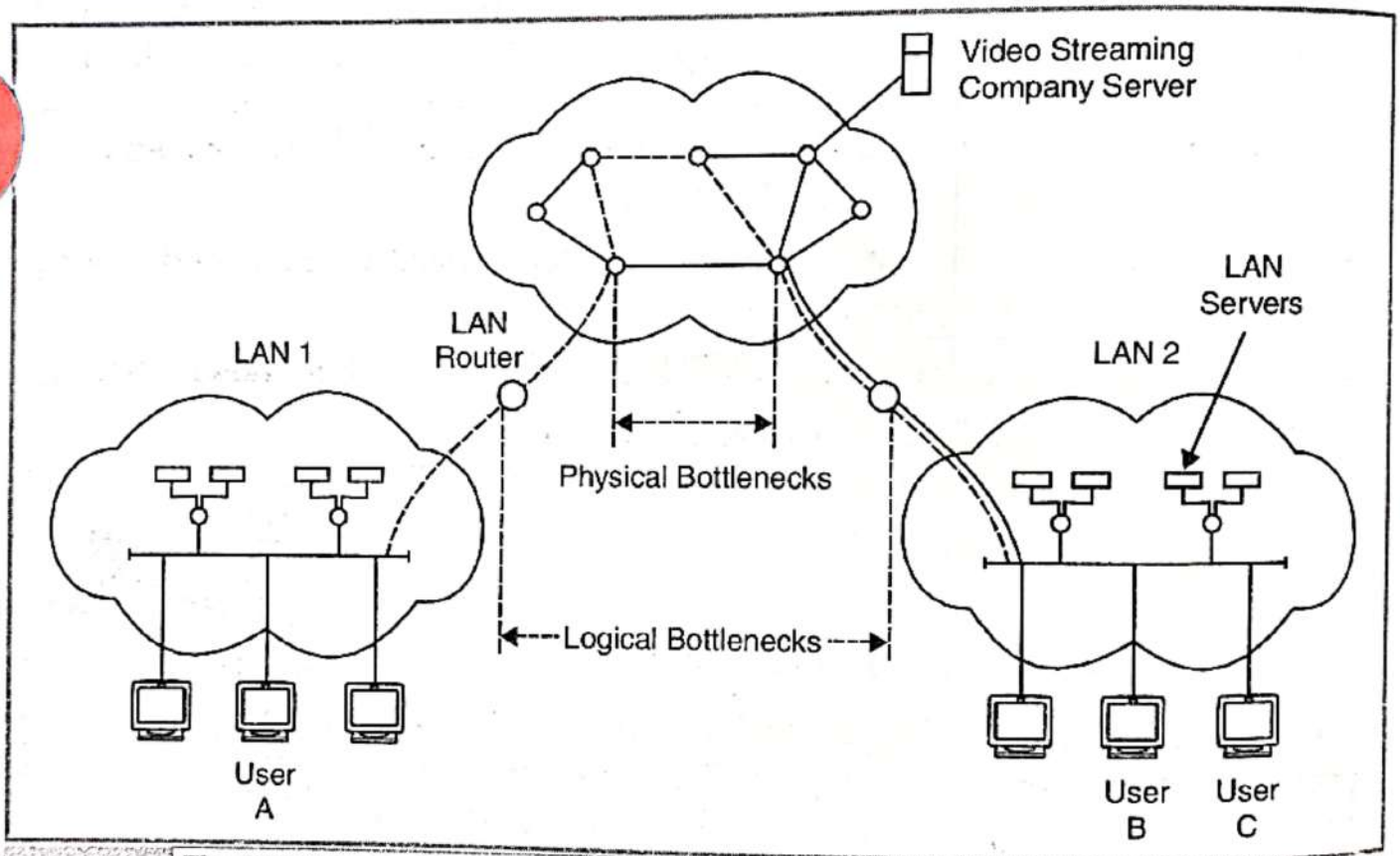


Fig. 6.25 Forms of bottleneck along a round-trip path between two users

- General methods of congestion control are either unidirectional or bidirectional. These two schemes are described next.

6.10.1 Unidirectional Congestion Control

- A network can be controlled unidirectionally through back-pressure signaling, transmission of choke packets, and traffic policing. Fig. 6.26 shows a network of eight routers : R1 through R8. These routers connect a variety of servicing companies : cellphones, satellite, residential, and company LANs.

- In such a configuration, congestion among routing nodes may occur at certain hours of the day.
- The first type of congestion control is achieved by generating a back-pressure signal between two routers.
- The back-pressure scheme is similar to fluid flow in pipes.
- When one end of a pipe is closed, the pressure propagates backward to slow the flow of water at the source.
- The same concept can be applied to networks. When a node becomes congested , it slows down the traffic on its incoming links until the congestion is relieved.
- For example, in the figure, router R4 senses overloading traffic and consequently sends signals in the form of back-pressure packets to router R3 and thereby to router R2, which is assumed to be the source of overwhelming the path. Packet flow can be controlled on a hop-by-hop basis.
- Back-pressure signaling is propagated backward to each node along the path until the source node is reached.
- Accordingly, the source node restricts its packet flow, thereby reducing the congestion.

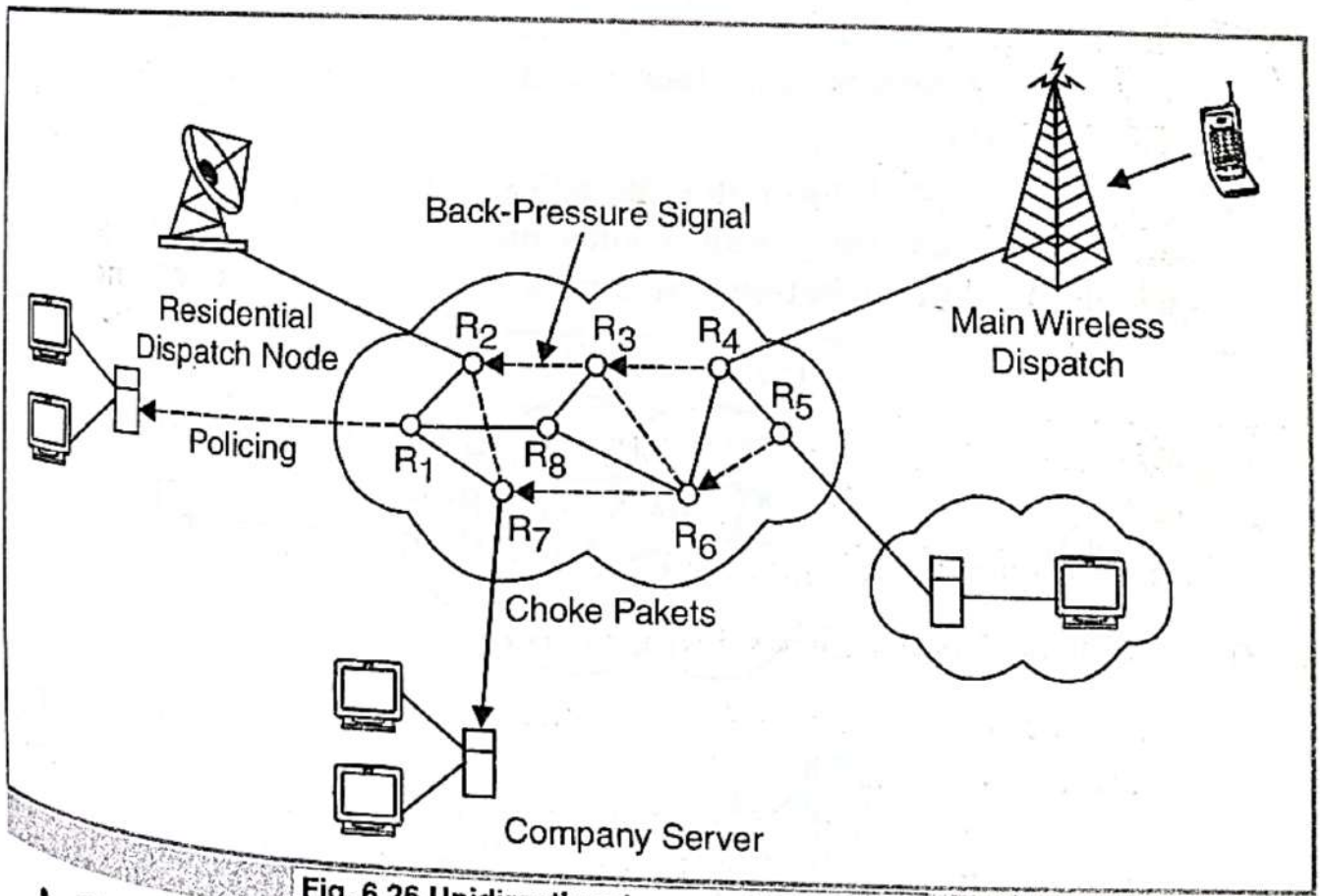


Fig. 6.26 Unidirectional congestion control

- Choke-packet transmission is another solution to congestion.

- In this scheme, choke packets are sent to the source node by a congested node to restrict the flow of packets from the source node.
- A router or even an end host can send these packets when it is near full capacity, in anticipation of a condition leading to congestion at the router.
- The choke packets are sent periodically until congestion is relieved.
- On receipt of the choke packets, the source host reduces its traffic-generation rate until it stops receiving them.
- The third method of congestion control is policing and is quite simple.
- An edge router, such as R1 in the figure, acts as a traffic police and directly monitors and controls its immediate connected consumers.
- In the Figure, R1 is policing a residential dispatch node from which traffic from a certain residential area is flowing into the network.

6.10.2 Bidirectional Congestion Control

Fig. 6.27 illustrates bidirectional congestion control, a host-based resource-allocation technique.

- The destination end host controls the rate at which it sends traffic, based on observable network conditions, such as delay and packet loss.
- If a source detects long delays and packet losses, it slows down its packet flow rate.
- All sources in the network adjust their packet-generation rate similarly; thus, congestion comes under control. Implicit signaling is widely used in packet-switched networks, such as the Internet.

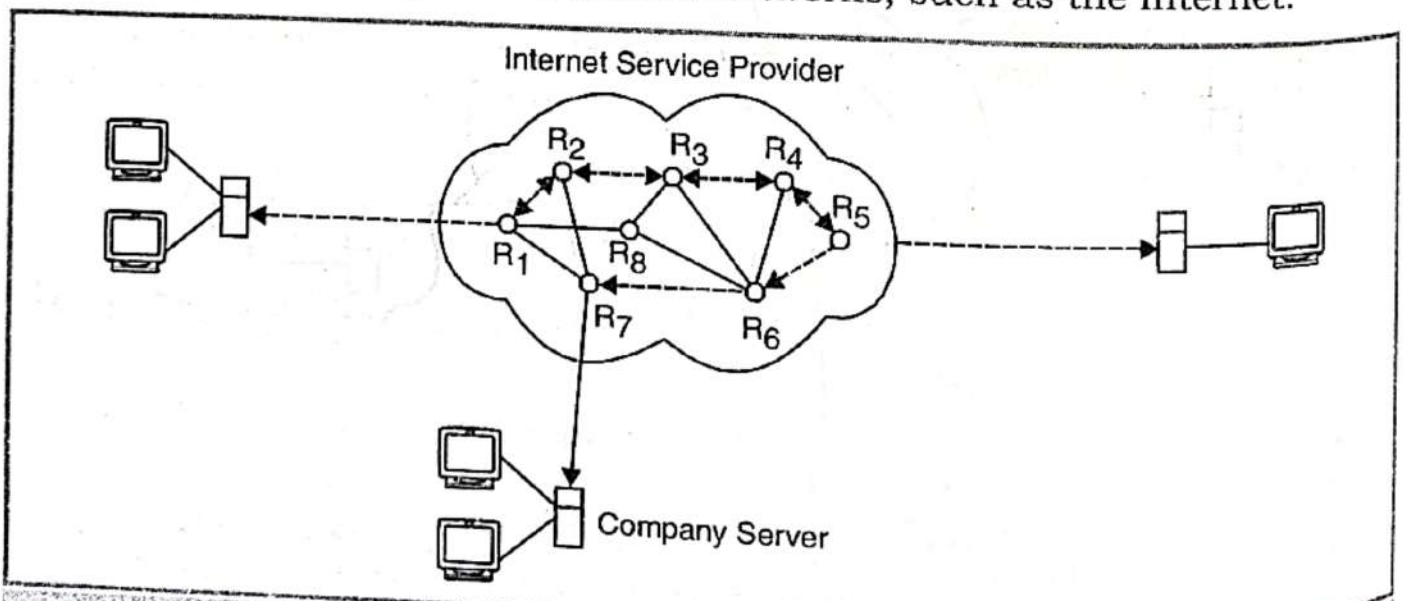


Fig. 6.27 Bidirectional congestion control

- In bidirectional signaling, network routing nodes alert the source of congested resources by setting bits in the header of a packet intended for the source.
- An alternative mechanism for signaling is to send control packets as choke packets to the source, alerting it of any congested resources in the network.
- The source then slows down its rate of packet flow.
- When it receives a packet, the source checks for the bit that indicates congestion.
- If the bit is set along the path of packet flow, the source slows down its sending rate.
- Bidirectional signaling can also be imposed by the window-based and rate-based congestion-control schemes.

6.10.3 Random Early Detection (RED)

- Random early detection (RED) avoids congestion by detecting and taking appropriate measures early.
- When packet queues in a router's buffer experience congestion, they discard all incoming packets that could not be kept in the buffer.
- This tail-drop policy leads to two serious problems : global synchronization of TCP sessions and prolonged congestion in the network.
- RED overcomes the disadvantages of the tail-drop policy in queues by randomly dropping the packets when the average queue size exceeds a given minimum threshold.
- From the statistical standpoint, when a queueing buffer is full, the policy of random packet drop is better than multiple-packet drop at once.
- RED works as a feedback mechanism to inform TCP sessions that the source anticipates congestion and must reduce its transmission rate.
- The packet-drop probability is calculated based on the weight allocation on its flow.
- For example, heavy flows experience a large number of dropped packets.
- The average queue size is computed, using an exponentially weighted moving average so that RED does not react to spontaneous transitions caused by bursty Internet traffic.

- When the average queue size exceeds the maximum threshold, all further incoming packets are discarded.

RED Setup at Routers

- With RED, a router continually monitors its own queue length and available buffer space.
- When the buffer space begins to fill up and the router detects the possibility of congestion, it notifies the source implicitly by dropping a few packets from the source.
- The source detects this through a time-out period or a duplicate ACK.
- Consequently, the router drops packets earlier than it has to and thus implicitly notifies the source to reduce its congestion window size.
- The "random" part of this method suggests that the router drops an arriving packet with some drop probability when the queue length exceeds a threshold.
- This scheme computes the average queue length, $E[N_q]$, recursively by

$$E[N_q] = (1 - \alpha)\, E\,[N_q] + \alpha\, N_i \qquad \ldots(6.1)$$

where N_i is the instantaneous queue length, and $0 < \alpha < 1$ is the weight factor.

- The average queue length is used as a measure of load.
- The Internet has bursty traffic, and the instantaneous queue length may not be an accurate measure of the queue length.
- RED sets minimum and maximum thresholds on the queue length, N_{min} and N_{max}, respectively.
- A router applies the following scheme for deciding whether to service or drop a new packet.
- If $E[N_q]$ N_{max}, any new arriving packet it dropped. If $E[N_q]N_{min}$, the packet is queued.
- If $N_{min} < E[N_q] < N_{max}$, the arriving packet is dropped with probability P given by

$$P = \frac{\delta}{1 - c\delta} \qquad \ldots(6.2)$$

where coefficient c is set by the router to determine how quickly it wants to reach a desired P.

In fact, c can be thought of as the number of arriving packets that have been queued.

We can them obtain from

$$= \frac{E[N_q]-N_{min}}{N_{max}-N_{min}} \quad ...(6.3)$$

- In essence, when the queue length is below the minimum threshold, the packet is admitted into the queue.
- Fig. 6.28 shows the variable setup in RED congestion avoidance. When the queue length is between the two thresholds, the packet-drop probability increases as the queue length increases.
- When the queue length is above the maximum threshold, the packet is always dropped. Also, shown in Equation (6.3), the packet drop probability depends on a variable that represents the number of arriving packets from a flow that has been queued.
- When the queue length increases, all that is needed is to drop one packet from the source.
- The source then halves its congestion window size.

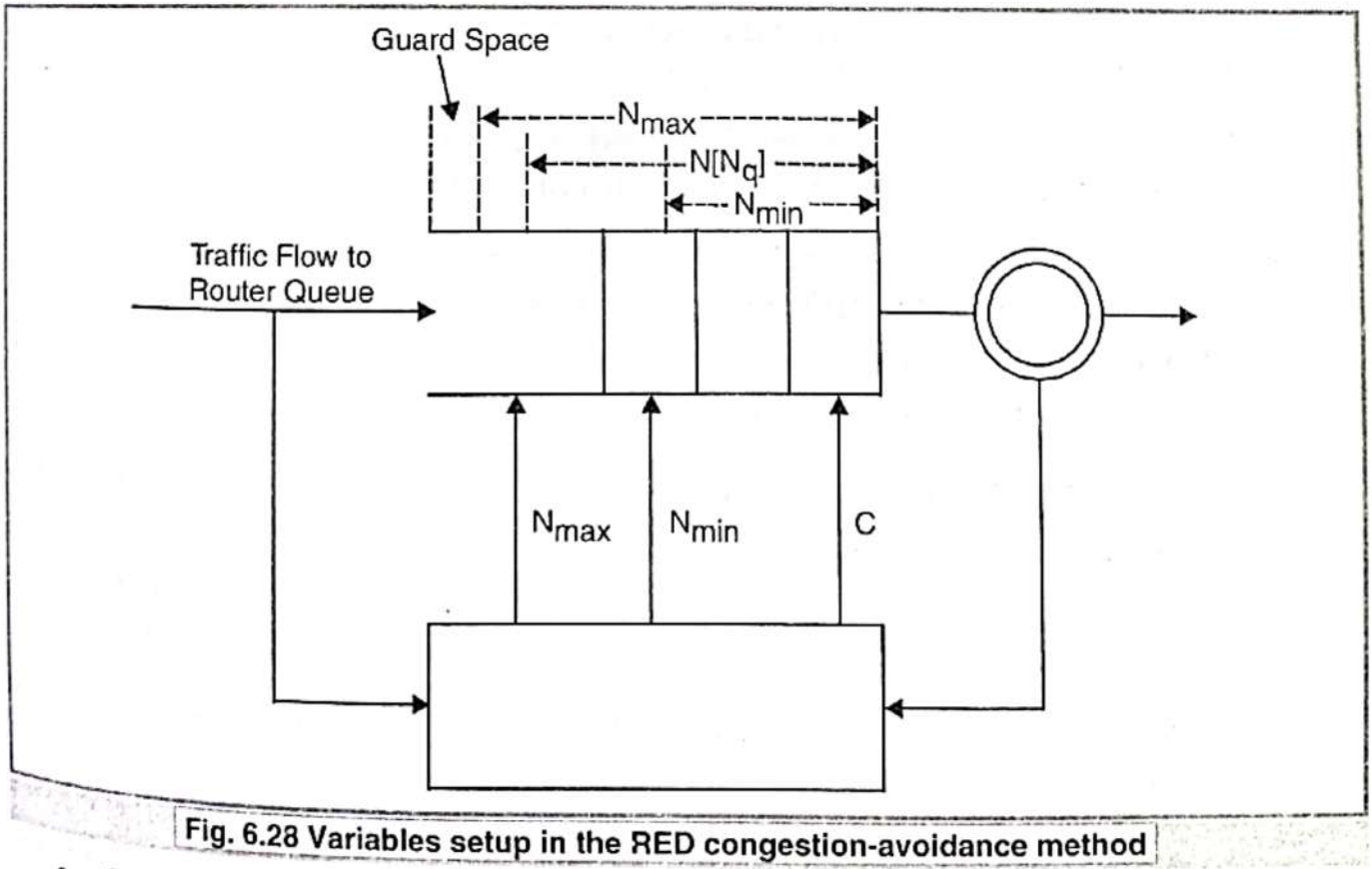


Fig. 6.28 Variables setup in the RED congestion-avoidance method

- Once a small number of packets are dropped, the associated sources reduce their congestion windows if the average queue length exceeds the minimum threshold, and therefore the traffic to the router drops.

With this method, any congestion is avoided by an early dropping of packets.

- RED also has a certain amount of fairness associated with it; for larger flows, more packets are dropped, since P for these flows could become large.
- One of the challenges in RED is to set the optimum values for N_{min}, N_{max}, and c.
- Typically, N_{min} has to be set large enough to keep the throughput at a reasonably high level but low enough to avoid congestion.
- In practice, for most networks on the Internet, the N_{max} is set to twice the minimum threshold value.
- Also, as shown by guard space in fig. 6.58, there has to be enough buffer space beyond N_{max}, as Internet traffic is bursty.

6.10.4 A Quick Estimation of Link Blocking

- A number of techniques can be used to evaluate a communication network's blocking probabilities.
- These techniques can vary according to accuracy and to network architectures.
- One of the most interesting and relatively simple approaches to calculating the level of blocking involves the use of Lee's probability graphs.
- Although Lee's technique requires approximations, it nonetheless provides reasonably accurate results.

Serial and Parallel Connection Rules

- Lee's method is based on two fundamental rules of serial and parallel connections. Each link is represented by its blocking probability, and the entire network of links is evaluated, based on blocking probabilities represented on each link, using one or both of the rules.
- This approach is easy to formulate, and the formula directly relates to the underlying network structures, without requiring any other detailed parameters.
- Thus, Lee's approach provides insight into the network structures, giving a remarkable solution for performance evaluations.
- Let p be the probability that a link is busy, or the percentage of link utilization.

- Thus, the probability that a link is idle is denoted by $q = 1-p$. Now, consider a simple case of two links in parallel to complete a connection with probabilities p_1 and p_2.
- The composite blocking probability, B_p, is the probability that both links are in use, or

$$B_p = p_1 p_2 \qquad ...(6.4)$$

- If these two links are in series to complete a connection, the blocking probability, B_s, is determined as 1 minus the probability that both links are available :

$$B_s = 1 - (1-p_1)(1-p_2) \qquad ...(6.4)$$

We can generalize the estimation of a network of links by two fundamental rules.

1. Rule 1 :

- For a parallel connection of links, the blocking probability is estimated by forming the product of the blocking probabilities for the subnetworks, as shown in Fig. 6.29.
- Let a source and a destination be connected in general through n links with probabilities p_1 through p_n, respectively, as shown in Fig. 6.29. Therefore, if the source and the destination are linked through parallel connections, the probability of blocking B_p is obtained from a product from as follows :

$$B_p = p_1 p_2 \cdots p_n \qquad ...(6.5)$$

2. Rule 2 :

- For a serial connection of links, the probability of blocking is estimated by forming the product of the probabilities of no blocking for the network.
- This method makes the assumption that the probability that a given link is busy is independent from link to link. Although this independence assumption is not strictly correct, the resulting estimates are sufficiently accurate.
- If links are in series, the probability of blocking is obtained from

$$B_s = 1 - (1-p_1)(1-p_2) \ldots (1-p_n) \qquad ... (6.6)$$

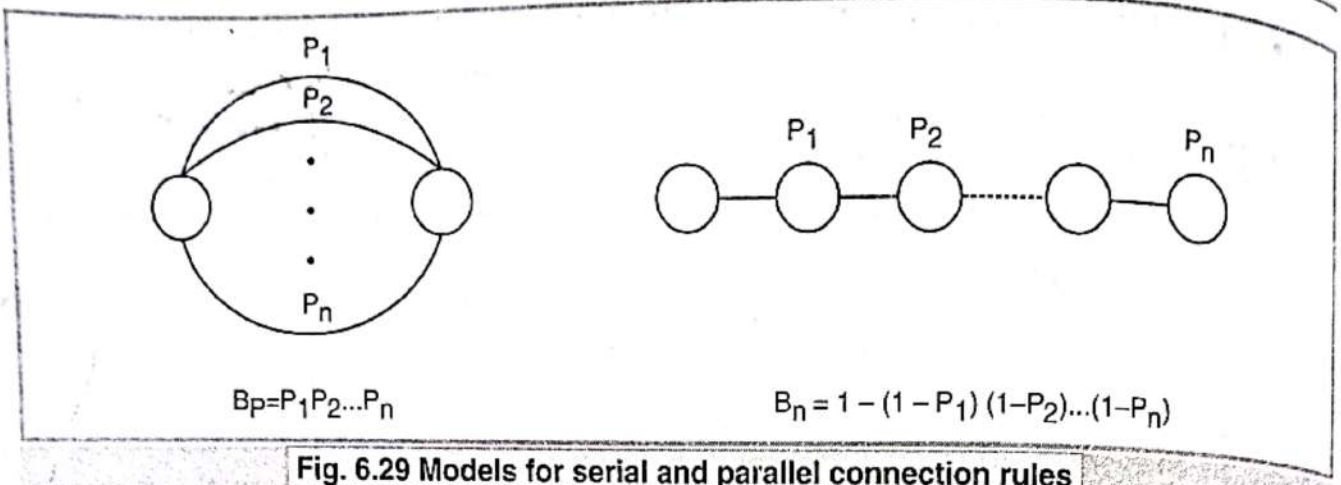


Fig. 6.29 Models for serial and parallel connection rules

Example 3. The network in Fig. 6.30 has six nodes, A through F, interconnected with ten links. The corresponding blocking probabilities of links, p_1 through p_{10}, are indicated on the corresponding links. For this network, find the overall blocking probability from A to F.

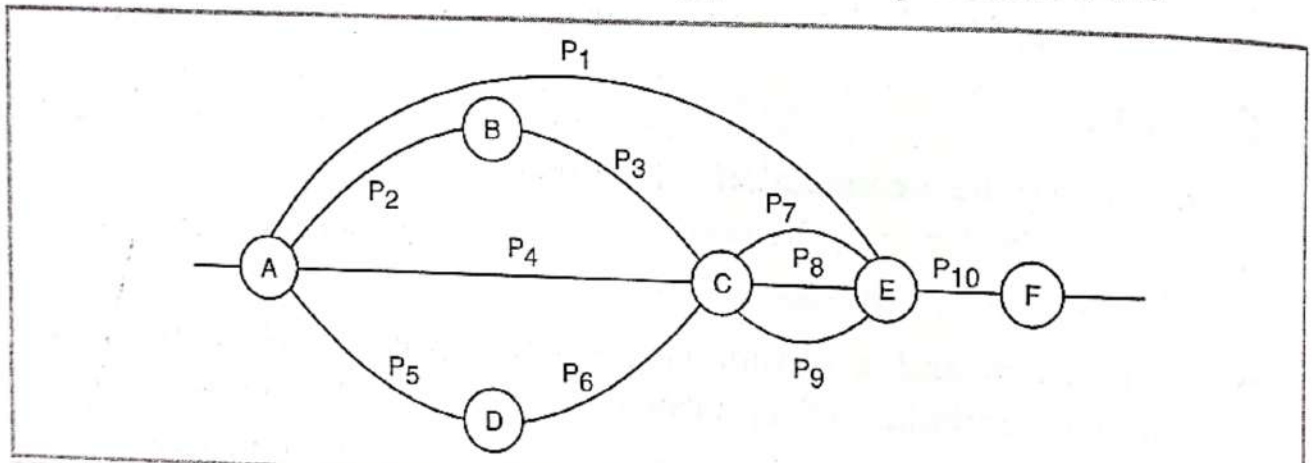


Fig. 6.30 : A network of links and their blocking probabilities

Solution. This network consists of a number of serial and parallel connections : for example,

$p_{ADC} = 1 - (1-p_5)(1-p_6)$, $p_{ABC} = 1 - (1-p_2)(1-p_3)$, and $p_{CE} = p_7 p_8 p_9$.

Thus : $B = p_{AF} = 1 - \{1-[1-(1- p_4 \cdot p_{ABC} \cdot p_{ADC})(1-p_{CE})]p_1\}(1-p_{10})$.

6.10.5 Leaky-Bucket Traffic Shaping

- This algorithm converts any turbulent incoming traffic into a smooth, regular stream of packets.
- Fig. 6.31 shows how this algorithm works.
- A leaky-bucket interface is connected between a packet transmitter and the network.
- No matter at what rate packets enter the traffic shaper, the outflow is regulated at a constant rate, much like flow of water from a leaky bucket.

- The implementation of a leaky-bucket algorithm is not difficult.

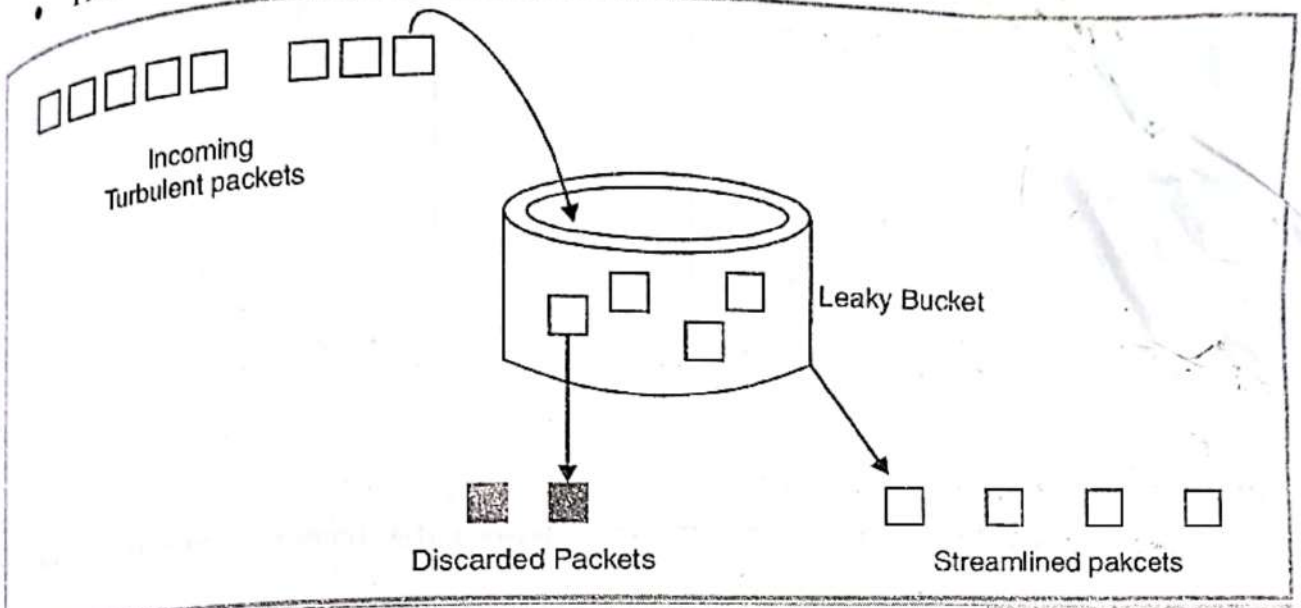


Fig. 6.31 The leaky-bucket traffic-shaping algorithm

- This scheme is a finite queue. When a packet arrives, the interface decides whether that packet should be queued or discarded, depending on the capacity of the buffer.
- The number of packets that leave the interface depends on the protocol.
- The packet-departure rate expresses the specified behaviour of traffic and makes the incoming bursts conform to this behaviour.
- Incoming packets are discarded once the bucket becomes full.
- This method directly restricts the maximum size of a burst coming into the system.
- Packets are transmitted as either fixed-size packets or variable-size packets. In the fixed-size packet environment, a packet is transmitted at each clock tick.
- In the variable-size packet environment, a fixed-sized block of a packet is transmitted.
- Thus, this algorithm is used for networks with variable-length packets and also equal-sized packet protocols, such as ATM.
- The leaky-buckey scheme is modeled by two main buffers, as shown in Fig. 6.32.
- One buffer forms a queue of incoming packets, and the other one receives authorizations.
- The leaky-bucket traffic-shaper algorithm is summarized as follows.

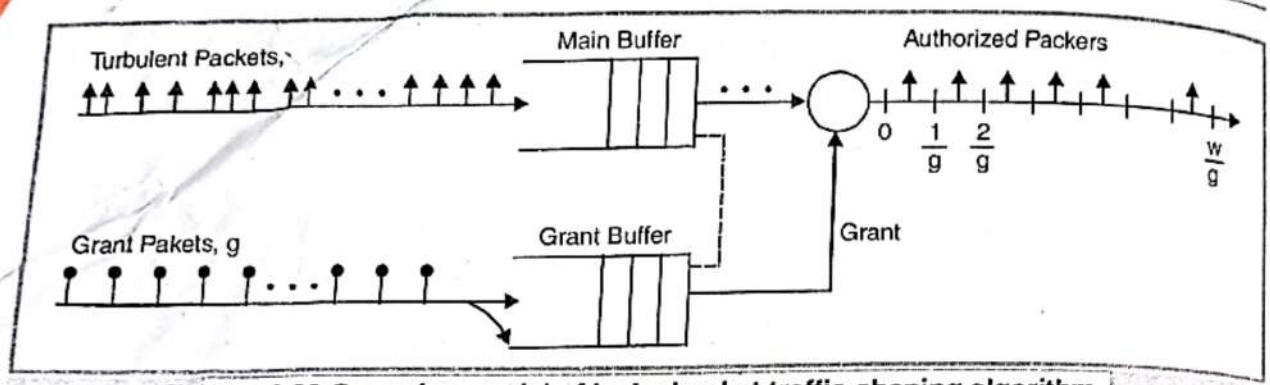


Fig. 6.32 Queueing model of leaky-bucket traffic-shaping algorithm.

Begin Leaky-Bucket Algorithm

1. Define for the Algorithm

   λ = rate at which packets with irregular rate arrive at the main buffer

   g = rate at which authorization grants arrive at the grant buffer

   ω = size of the grant buffer and can be dynamically adjusted

2. Every 1/g seconds, a grant arrvies.
3. Over each period of 1/g seconds, i grants can be assigned to the first i incoming packets, where i < ω, and packets exit from the queue one at a time every 1/g seconds, totaling i/g seconds.
4. If more than ω packets are in the main buffer, only the first ω packets are assigned grants at each window time of 1/g, and the rest remain in the main queue to be examined in the next 1/g interval.
5. If no grant is in the grant buffer, packets start to be queued.

- With this model, ω is the packet size.
- The bucket in this case is the size of the window that the grant buffer opens to allow ω packets from the main buffer to pass.
- This window size can be adjusted, depending on the rate of traffic.
- If ω is too small, the highly turbulent traffic is delayed by waiting for grants to become available.
- If ω is too large, long turbulent streams of packets are allowed into the network.
- Consequently, with the leaky-bucket scheme, it would be best for a node to dynamically change the window size (bucket size) as needed.
- The dynamic change of grant rates in high-speed networks may, however, cause additional delay through the required feedback mechanism.

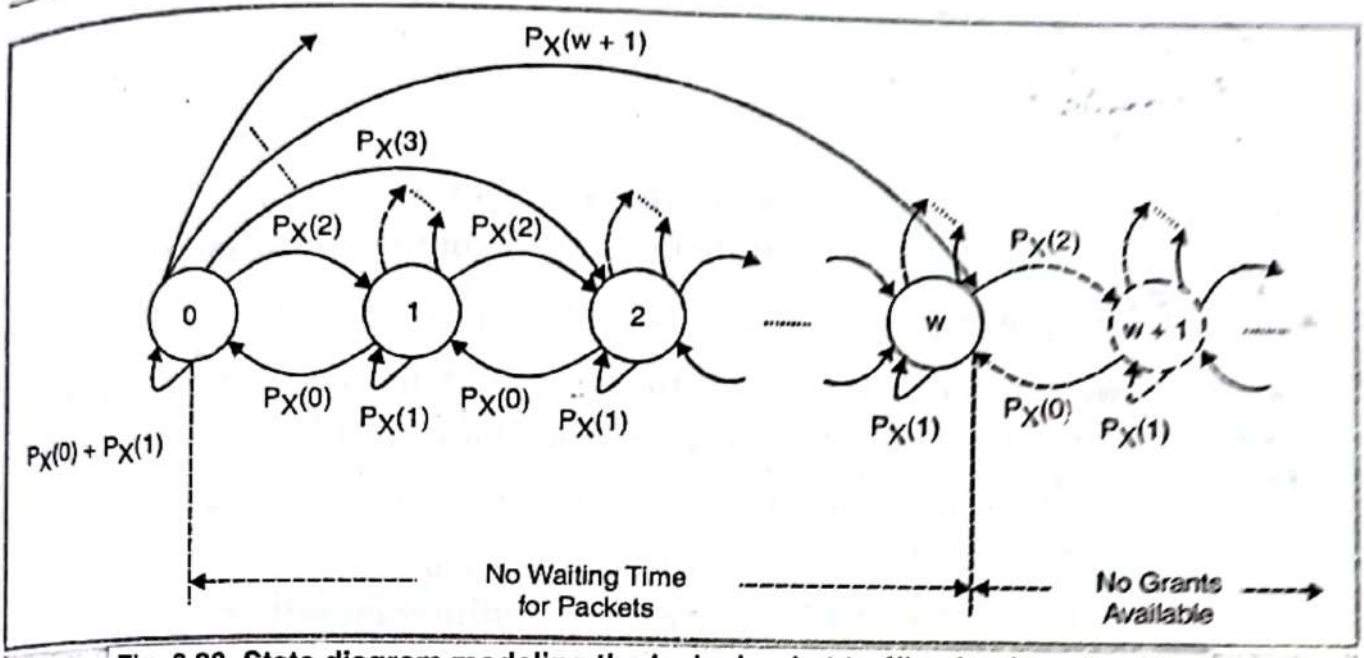


Fig. 6.33. State diagram modeling the leaky-bucket traffic-shaping algorithm

- Fig. 6.33 shows the Markov chain state diagram depicting the activity of grant generation in time.
- At the main buffer, the mean time between two consecutive packets is $1/\lambda$.
- At the grant buffer, a grant arrives at rate g. Hence, every $1/g$ seconds, a grant is issued to the main buffer on times 0, 1,/g, 2/g, ... If the grant buffer contains ω grants, it discards any new arriving grants.
- A state $i \in \{0, 1, ...\omega\}$ of the Markov chain refers to the situation in which i grants are allocated to i packets in the main buffer, and thus $\omega - i$ grants are left available.
- In this case, the i packets with allocated grants are released one at a time every $1/g$ seconds.
- When the packet flow is slow, the grant queue reaches its full capacity, and thus grant $(\omega+1)$ and beyond are discarded. We define the following main variables for delay analysis :
  - Px(x) = The probability of x packet arrivals in $1/g$ seconds
  - Pi = The probability that a grant has arrived or that the Markov chain is in state i
  - Pji = The transition probability from any state j to a state i on the Markov chain
- As shown in Fig.6.32, the chain starts at state 0, implying that ω grants are available in the grant buffer and that no packet is in the main buffer.

- In this state, the first arriving packet to the main buffer with probability $P_x(1)$ gets a grant while a new grant arrives in the same period.
- This creates no changes in the state of the chain. Therefore, the transition probability at state 0 has two components, as follows:

$$P_{00} = P_x(0) + P_x(1) \quad (1)$$

- This means that P_{00} is equal to the probability that no packet or only one packet arrives ($P_x(0) + P_x(1)$). As long as $i > 1$, state 0 can be connected to any state $i < \omega$, inducing the property of the queueing system in Fig. 30 by

$$P_{0i} = P_x(i+1) \text{ for } i \geq 1 \quad (2)$$

- For example, two new packets arriving during a period 1/g with probability Px(2) change state 0 to state 1 so that one of the two packets is allocated a grant, and thus the chain moves to state 1 from state 0, and the second packet is allocated the new arriving grant during that period of time.
- The remaining transition probabilities are derived from

$$P_{ji} = \begin{cases} P_x = (i-j+1) \text{ for } 1 \leq j \leq i+1 \\ 0 \qquad\qquad \text{for } j > i+1 \end{cases} \quad (3)$$

- Now, the global balance equations can be formed. We are particularly interested in the probability of any state i denoted by P_i. The probability that the chain is in state 0, P_0, implying that no grant has arrived, is the sum of incoming transitions :

$$P_0 = P_x(0) P_1 + [P_x(0) + P_x(1)] P_0 \quad (4)$$

For P_1, we can write $P_1 = P_x(2) P_0 + P_x(1) P_1 + P_x(0) P_2$ (5)

For all other states, the probability of the state can be derived from the following generic expression :

$$P_i = \sum_{j=0}^{i+1} P_x(i-j+1) P_j \quad \text{for } i \geq 1 \quad \ldots(6)$$

The set of equations generated from Equation (6.6) can be recursively solved.

Knowing the probability of each state at this point, we can use Little's

formula to estimate the average waiting period to obtain a grant for a packet, E(T), as follows :

$$E[T] = \frac{\sum_{i=w+1}^{\infty} (i-w)P_i}{g} \quad ...(7)$$

- It is also interesting to note that the state of the Markov chain can turn into "queueing of packets" at any time, as shown by dashed lines in Fig. 6.33.
- For example at state 0, if more than $\omega+1$ packets arrive during $1/g$, the state of the chain can change to any state after ω, depending on the number of arriving packets during $1/g$. If this happens, the system still assigns $\omega + 1$ grants to the first $\omega + 1$ packets and assigns no grant to the remaining packets.
- The remaining packets stay pending to receive grants.

SUMMARY

- The services of network layer are provided to transport layer which can be either connection oriented or connection less. The OSI supports both type of services. In connection oriented services three phases of connection establishment, data transfer and connection release are used. In connectionless service only data transfer and receiving primitives are available.
- The main functions of network layer are network connection establishment, routing, multiplexing, error detection, segmentation and blocking of N-PDU.
- A routing algorithm must have simplicity, stability fairness, correctness, Robustness and optimality. Many routing algorithms are known. Most widely used algorithm is shortest path routing. A single routing machine computers all the routes and down loads them on IMP's in centralized routing. The decision is made by each IMP, itself based on local traffic conditions in isolated routing. The distributed routing lies between two and in it each IMP makes local routing decisions by exchanging information's to its neighbours. In large networks the hierarchical routing is used.
- If too many packets are present in a subnet it causes congestion. It drops the throughput and sometimes it may lead to dead locks. Various congestion control methods are discarding packets, pre allocating buffers, limiting number of packets etc. Various algorithms

are known to avoid dead locks. Two types of dead locks are direct store and forward deadlock and Indirect store and forward dead lock.

- Internetworking means connecting two or more networks together. The connections are possible using bridges, routers or gateways. Bridges operate at data link layer, gateways at application layer and routers operate at network layer. Two types of bridges are spanning tree bridge and source routing bridge.
- Two types of interconnections available in network layer are virtual circuits and internet data grams (virtual call). In virtual circuit congestion is avoided and functions are simple but in datagram congestion can occur though virtual circuit approach is inflexible and datagram approach is flexible.

EXERCISE

I. FILL UP THE BLANKS

1. The third layer of OSI model is.....
2. The network layer is responsible for.....
3. The network layer is known as inactive layer because of..... in LAN's.
4. Two types of connections provided by a sub network are..... and.....
5. Two major classes of routing algorithms are..... and.....
6. Three types of adaptive algorithms are.....,..... and.....
7. The presence of too many packets on a subnet is known as.....
8. The ultimate congestion is known as.....
9. Two or more networks, interconnected is known as.....
10. The device connecting two similar LAN's is.....
11. The devices connecting two dissimilar LAN's are..... and.....
12. Gate ways operate at..... layer.
13. Routers operate at..... layer.
14. Bridges operate at..... sub layer of..... layer.
15. Two modes of network services are..... and.....
16. The network layer provides its services to..... layer.

II. VERY SHORT ANSWER TYPE QUESTIONS

1. What do you mean by fixed connection ?
2. Name any two switched networks ?
3. Define switched connections ?
4. What is a routing Algorithm ?
5. How many types of routing algorithms are there ?
6. What is shortest path routing ?
7. What do you mean by broadcast routing ?
8. What are bridges ?
9. Define Routers ?
10. What are Gateways ?
11. What do you mean by congestion ?
12. What is a deadlock ?
13. What is an internetworking ?
14. What do you mean by isolated routing ?
15. What is the main function of network layer ?
16. What is a hierarchical routing ?

III. SHORT ANSWER TYPE QUESTIONS

1. What are the functions of network layer ? Discuss in brief.
2. Discuss in brief the service primitives of network layer ?
3. Explain connection mode network services ?
4. Explain connection less mode network services ?
5. What is a routing algorithm ? What are its characteristics properties ?
6. Explain shortest path routing ?
7. Discuss multipath routing in brief ?
8. Explain isolated Routing ?
9. Explain centralized routing in brief ?
10. Discuss distributed routing in detail ?
11. Explain congestion and deadlock ?

12. What do you mean by internetworking ? What are the hardware requirement for internetworking ?

IV. LONG ANSWER TYPE QUESTIONS

1. What do you mean by network layer ? Explain its functions in detail.
2. What do you mean by network services ? Explain various types of network services in brief.
3. Explain various network service primitives ?
4. Explain routing algorithms along with their types and properties in brief ?
5. Explain adaptive routing methods in brief ?
6. Explain congestion and its control methods in detail ?
7. Write short notes on
   (a) Internetworking.
   (b) Bridges
   (c) Routers
   (d) Gateways
8. What is the need for network layer ? Discuss in brief.

ANSWERS

I. FILL UP THE BLANKS

1. Network layer 2. routing packets across the network, 3. its lack of functionality 4. fixed, switched 5. adaptive, non adaptive 6. isolated, centralized, distributed 7. congestion 8. dead lock 9. internet work 10. bridges 11. routers, gateways 12. application 13. network 14. MAC, data link 15. connection oriented, connectionless 16. transport

□□□□□