COMPUTER COMMUNICATION AND NETWORKS

BCA III SEM(NEP)

NOTES

Prepared By

Mr. Prabhu Kichadi, BE, MTECH

9880437187

UNIT – IV

Prabhu Kichadi 9880437187

The Network Layer: Network Layer Design Issues, Routing Algorithms – Flooding, Distance Vector Routing, Hierarchical Routing, Link State Routing, Congestion, Control Algorithms – Leaky Bucket, Token Bucket Algorithm, Admission Control, Hop by Hop Choke Packets.

The Network Layer: Network layer is majorly focused on getting packets from the source to the destination, routing error handling and congestion control.

Various functions of network layer.

- **Addressing:**
  Maintains the address at the frame header of both source and destination and performs addressing to detect various devices in network.
- **Packeting:**
  This is performed by Internet Protocol. The network layer converts the packets from its upper layer.
- **Routing:**
  It is the most important functionality. The network layer chooses the most relevant and best path for the data transmission from source to destination.
- **Inter-networking:**
  It works to deliver a logical connection across multiple devices.

Network layer design issues:

1. Store and Forward packet switching: The host sends the packet to the nearest router. This packet is stored there until it has fully arrived once the link is fully processed by verifying the checksum then it is forwarded to the next router till it reaches the destination. This mechanism is called "Store and Forward packet switching.

2. Services provided to Transport Layer: Through the network/transport layer interface, the network layer transfers it's services to the transport layer.

Based on the connections there are 2 types of services provided :

Connectionless – The routing and insertion of packets into subnet is done individually. No added setup is required.

Connection-Oriented – Subnet must offer reliable service and all the packets must be transmitted over a single route.

3. Implementation of Connectionless Service:

Packet is termed as "datagrams" and corresponding subnet as "datagram subnets". When the message size that has to be transmitted is 4 times the size of the packet, then the network layer divides into 4 packets and transmits each packet to router via. a few protocols. Each data packet has destination address and is routed independently irrespective of the packets.

4. Implementation of Connection Oriented service:

To use a connection-oriented service, first we establish a connection, use it and then release it. In connection-oriented services, the data packets are delivered to the receiver in the same order in which they have been sent by the sender.

Routing algorithms:

- In order to transfer the packets from source to the destination, the network layer must determine the best route through which packets can be transmitted.
- Whether the network layer provides datagram service or virtual circuit service, the main job of the network layer is to provide the best route. The routing protocol provides this job.
- The routing protocol is a routing algorithm that provides the best path from the source to the destination. The best path is the path that has the "least-cost path" from source to the destination.
- Routing is the process of forwarding the packets from source to the destination but the best route to send the packets is determined by the routing algorithm.

Classification of a Routing algorithm

The Routing algorithm is divided into two categories:

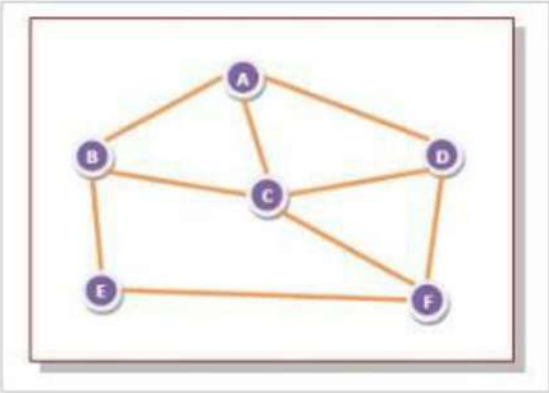
- **Adaptive Routing algorithm(Dynamic Routing algorithm)**
  - An adaptive routing algorithm is also known as dynamic routing algorithm.
  - This algorithm makes the routing decisions based on the topology and network traffic.
  - The main parameters related to this algorithm are hop count, distance, and estimated transit time.
- **Non-adaptive Routing algorithm(Static Routing algorithm)**
  - Non Adaptive routing algorithm is also known as a static routing algorithm.
  - When booting up the network, the routing information stores to the routers.
  - Non-Adaptive routing algorithms do not take the routing decision based on the network topology or network traffic.
  - Example - Flooding

Flooding:

In case of flooding, every incoming packet is sent to all the outgoing links except the one from it has been reached. The disadvantage of flooding is that node may contain several copies of a particular packet.

Flooding is a non-adaptive routing technique following this simple method: when a data packet arrives at a router, it is sent to all the outgoing links except the one it has arrived on.

For example, let us consider the network in the figure, having six routers that are connected through transmission lines.



Using flooding technique –

- An incoming packet to A, will be sent to B, C and D.
- B will send the packet to C and E.
- C will send the packet to B, D and F.
- D will send the packet to C and F.
- E will send the packet to F.
- F will send the packet to C and E.

Advantages of Flooding

- It is very simple to setup and implement, since a router may know only its neighbours.
- It is extremely robust. Even in case of malfunctioning of a large number routers, the packets find a way to reach the destination.

Limitations of Flooding

- Flooding tends to create an infinite number of duplicate data packets, unless some measures are adopted to damp packet generation.
- It is wasteful if a single destination needs the packet, since it delivers the data packet to all nodes irrespective of the destination.

Distance Vector Routing Algorithm:

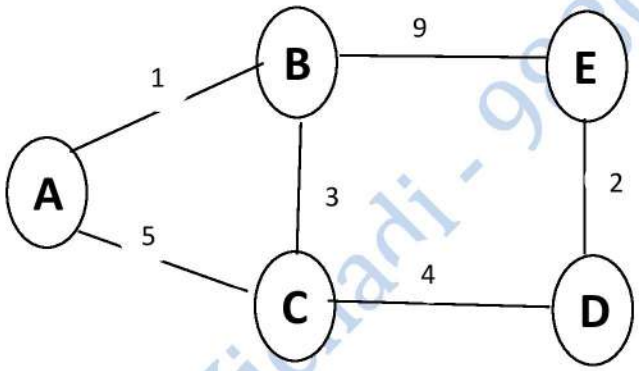
- It is a dynamic(adaptive) iterative routing algorithm.
- In this algorithm every node maintains updated routing table, ie the shortest and optimized route to destination including the next hop node.

- For any source x to any destination, the minimum cost path is given by **Bellman ford algorithm** is

$$D_{(x)}y = \min \{ c(x,v) + d_{(v)}Y\}$$

- In this algorithm every node must maintains a routing table(vector) information as given below,
- In this algorithm, the **cost and minimum no of intermediate nodes must be considered.**

Destination	Cost	next-hop



Use Bellman ford algorithm and calculate the minimum cost and less no intermediate node route.

i. Minimum Cost from A to B:

A->B : 1

A->C->B: 8

A->C->D->E->B:20

ii. Minimum Cost from A to C:

A->C : 5

A->B->C: 4

A->B->E->D->C:16

Prof Prabhu Kichadi - 9880437187

iii. Minimum Cost from A to D:

A->C->D : 9

A->B->C->D: 8

A->B->E->D:12

iii. Minimum Cost from A to E:

A->B->E : 10

A->B->C->D->E:10

A->C->D->E:11

Routing table for hop 'A':

Destination	Cost	next-hop
B	1	A
C	4	B
D	8	B
E	10	B

Prof. Prabhu Kichadi - 9880437187

Link State Routing Algorithm:

It is used to find the shortest/optimum path from one node to every node.
Two phases involved,

Phase 1: Reliable Flooding:

Link state information is flooded across the network.
-each node knows the cost of its neighbor.
- Each node knows the entire graph.
- Link state information is finding by every node if its directly connected to the neighbor and the cost.

Phase 2: Route Calculation(Shortest Path Tree):

- Each node uses dijsktra algorithm the optimal path. It uses dijsktra algorithm to find the Least Cost Tree,

```
For any node X, root node,
{
        If Y is the root node,
                Then d(x,y) = 0 (distance)
        Else if Y is neighbor node
                Then d(x,y) = c(x,y)
        Else
                D(x,y) = infinity
}
Repeat
{
        Find a node with d(x) minimum from root node.
                For (every node x, which is a neighbor) using
        Bellman Ford Algorithm,
                D(x,y) = min{c(x,v) + d(x,y)}
}
```

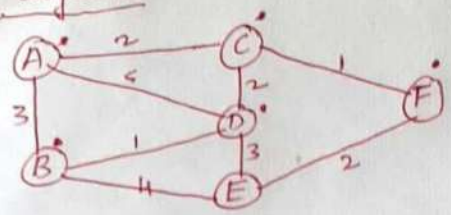
Example with solution:

Prof. Prabhu Kichadi - 9880437187

For a given network,

Link state algorithm

→



(1) Reliable Flooding

→ Every node maintains a link state (neighbour cost) information about it's neighbour and sends it as flooding.

→ That is, For nodes

A,

node	cost
B	3
C	2

C,

node	cost
A	2
D	2
F	1

B,

node	cost
A	3
D	1
E	4

D,

node	cost
A	5
B	1
C	2
E	3

E,

node	cost
B	4
D	3
F	2

F,

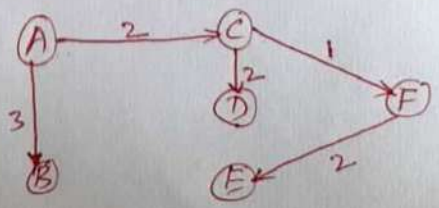
node	cost
C	1
E	2

(2) Find the minimum cost tree (shortest path tree) using dijkstra's & bellman ford algorithm, the table is

source 'A'

Iteration	Tree	B	C	D	E	F
Initial	{A}	3	[2]	5	∞	∞
1	{A, C}	[3]	[2]	4	∞	3
2	{A, C, B}	[3]	[2]	4	7	[3]
3	{A, C, B, F}	-	-	[4]	5	-
4	{A, C, B, F, D}	-	-	-	[5]	-
5	{A, C, B, F, D, E}	-	-	-	-	-

The least cost tree is

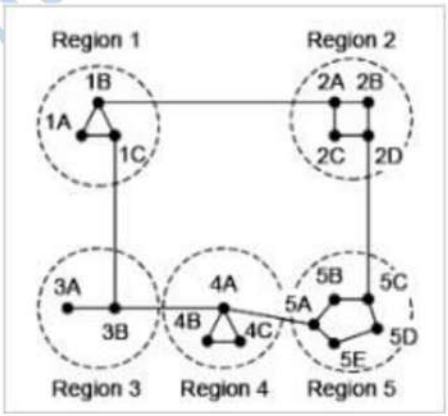


Hierarchical Routing Algorithm:

- When the network size grows, the number of routers in the network increases. Consequently, the size of the routing table increases. Hence routers can't handle network traffic as efficiently.
- We use hierarchical routing to overcome this problem.
- It reduces the size of the routing tables.
- Provides better scalability.
- Hierarchical Routing : defines the network routers in terms of regions/clusters based on locations, topology and number of routers and allocates a leader router for each region and levels.
- It uses divide and conquer strategy, may be in terms of levels 0, 1 & 2.

- In hierarchical routing, the routers are divided into regions. Each router has complete details about how to route packets to destinations within its own region. But it does not have any idea about the internal structure of other regions.
- In hierarchical routing, routers are classified in groups called regions. Each router has information about the routers in its own region and it has no information about routers in other regions. So, routers save one record in their table for every other region.

Example



Let see the full routing table for router 1A which has 17 entries, as shown below –

Prof. Prabhu Kichadi - 9880437187

Full Table for 1A

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

When routing is done hierarchically then there will be only 7 entries as shown below –

Hierarchical Table for 1A

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

Explanation

Step 1 – For example, the best path from 1A to 5C is via region 2, but hierarchical routing of all traffic to region 5 goes via region 3 as it is better for most of the other destinations of region 5.

Step 2 – Consider a subnet of 720 routers. If no hierarchy is used, each router will have 720 entries in its routing table.

Step 3 – Now if the subnet is partitioned into 24 regions of 30 routers each, then each router will require 30 local entries and 23 remote entries for a total of 53 entries.

Congestion Control Algorithms:

Congestion: Network Congestion occurs in cases of traffic overloading when a link or network node is handling data in excess of its capacity.

Congestion Control is a mechanism that controls the entry of data packets into the network, enabling a better use of a shared network infrastructure and avoiding congestive collapse.

As far as the end-user is concerned, network congestion feels like slow response times or a "network slow down." When the internet, the WiFi, or even the computer itself just "feels slow," that is often the result of network congestion. However, there's more to network traffic congestion than this.

The factors that affect congestion are:

1. Bandwidth Usage

Bandwidth is among the most common causes of network congestion. Bandwidth refers to the maximum rate that data can move along a path, or the total capacity of that path.

2. Latency

Latency is the time it takes a data packet to travel from point A to point B.

3. Collisions

Packet collisions can be caused by poor cabling or bad equipment, and can produce a serious situation, forcing all packets to stop and wait for a clear network to retransmit.

Leaky Bucket:

The leaky bucket algorithm is ideal for smoothing out bursty traffic. Just like a hole at the bottom of a water bucket leaks water out at a fixed rate, the leaky bucket algorithm does the same with network traffic.

Bursty chunks of traffic are stored in a "bucket" with a "hole" and sent out at a controlled, average rate.

The hole represents the network's commitment to a particular bandwidth. The leaky bucket shapes the incoming traffic to ensure it conforms to the commitment. Thus, regardless of how much data traffic enters the bucket, it always leaves at a constant output rate (the commitment).

This mechanism regulates the packet flow in the network and helps to prevent congestion that leads to performance deterioration and traffic delays.

Leaky bucket example.

- Suppose data enters the network from various sources at different speeds. Consider one bursty source that sends data at 20 Mbps for 2 seconds for a total of 40 Mbps. Then it is silent, sending no data for 5 seconds. Then it again transmits data at a rate of 10 Mbps for 3 seconds, thus sending a total of 30 Mbps. So, in a time span of 10 seconds the source sends 70 Mb data.

- However, the network has only committed a bandwidth of 5 Mbps for this source. Therefore, it uses the leaky bucket algorithm to output traffic at the rate of 5 Mbps during the same time period of 10 seconds, which smooths out the network traffic.

Implementing the leaky bucket algorithm with a FIFO queue

- FIFO queuing means that the first packet that arrives at a router is also the first to be transmitted. Furthermore, if a packet arrives and the queue -- also known as the buffer space -- is full, it is discarded by the router.

- A FIFO queue can be used to implement a leaky bucket algorithm in a network. The queue holds the packets and doesn't allow them to pass if they exceed the bandwidth committed by the network for a source. If the traffic consists of variable-length packets, the output rate is fixed based on the commitment in bytes or bits.

- However, if the traffic consists of fixed-size packets, the algorithm will drop the packets arriving at the tail end of the FIFO. This phenomenon, known as a "tail drop," happens regardless of the packet's importance or which flow it belongs to.

Applications of leaky bucket algorithm:

The algorithm is also implemented to prevent congestion in telecommunications networks. When subscribers sign a contract with a telecom provider, they can use a specific bandwidth within a specified period (per day, per month, etc.) If the subscriber uses more bandwidth than is allocated to them, excess packets will spill out of the bucket.

Prof. Prabhu Kichadi 9880437187

Token Bucket Algorithm:

The leaky bucket algorithm enforces output patterns at the average rate, no matter how busy the traffic is. So, to deal with the more traffic, we need a flexible algorithm so that the data is not lost. One such approach is the token bucket algorithm.

Algorithm:

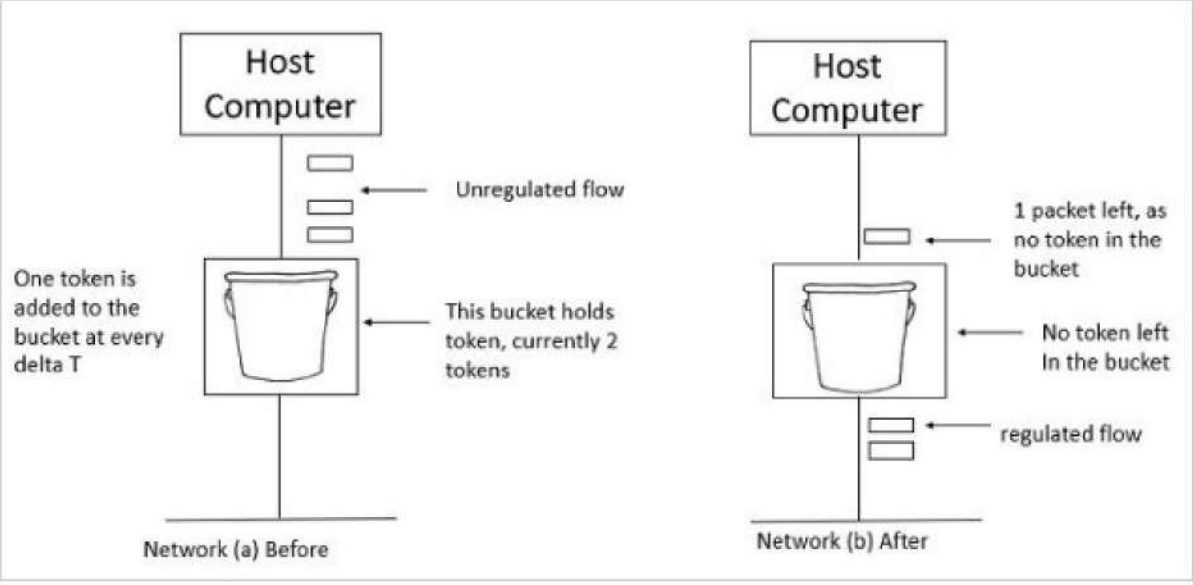
Step 1 – In regular intervals tokens are thrown into the bucket f.

Step 2 – The bucket has a maximum capacity f.

Step 3 – If the packet is ready, then a token is removed from the bucket, and the packet is sent.

Step 4 – Suppose, if there is no token in the bucket, the packet cannot be sent.

Example:



In figure (a) the bucket holds two tokens, and three packets are waiting to be sent out of the interface.

In Figure (b) two packets have been sent out by consuming two tokens, and 1 packet is still left.

When compared to Leaky bucket the token bucket algorithm is less restrictive that means it allows more traffic. The limit of busyness is restricted by the number of tokens available in the bucket at a particular instant of time.

The implementation of the token bucket algorithm is easy – a variable is used to count the tokens. For every t seconds the counter is incremented and then it is decremented whenever a packet is sent. When the counter reaches zero, no further packet is sent out.

LEAKY BUCKET ALGORITHM

- We have already seen one way to limit the amount of data an application sends:the sliding window, which uses one parameter to limit how much data is in transit at any given time, which indirectly limits the rate.
- Now we will look at a more general way to characterize traffic, with the leakybucket and token bucket algorithms.
- The formulations are slightly different but give an equivalent result.
- Try to imagine a bucket with a small hole in the bottom, as illustrated in Fig. 5- 28(b). No matter the rate at which water enters the bucket, the outflow is at constant rate, R, when there is any water in the bucket and zero when the bucketis empty.
- Also, once the bucket is full to capacity B, any additional water entering it spillsover the sides and is lost.

Prof. Prabhu Kichadi

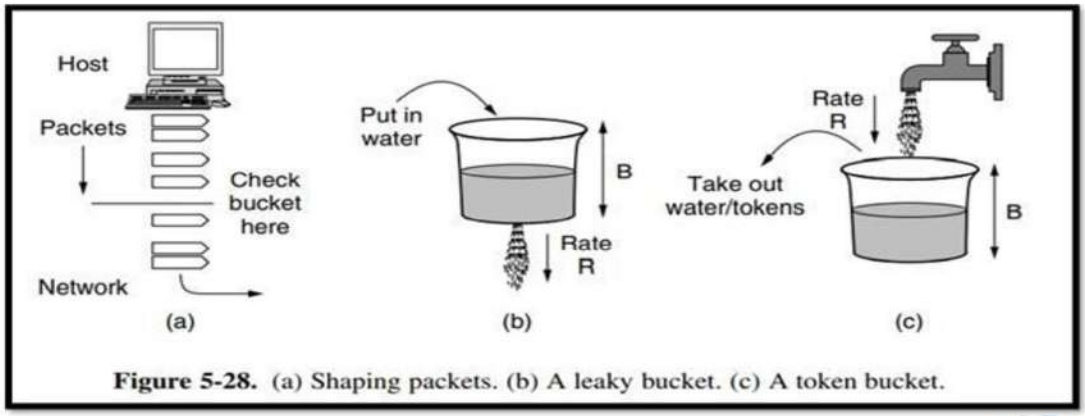


Figure 5-28. (a) Shaping packets. (b) A leaky bucket. (c) A token bucket.

- This bucket can be used to shape or police packets entering the network, as shown in Fig. 5-28(a).
- Conceptually, each host is connected to the network by an interface containing a leaky bucket.
- To send a packet into the network, it must be possible to put more water into the bucket. If a packet arrives when the bucket is full, the packet must either be queued until enough water leaks out to hold it or be discarded.
- The former might happen at a host shaping its traffic for the network as part of the operating system.
- The latter might happen in hardware at a provider network interface that is policing traffic entering the network.
- This technique was proposed by Turner (1986) and is called the leaky bucket algorithm.
- The tap is running at rate R and the bucket has a capacity of B, as before.
- Now, to send a packet we must be able to take water, or tokens, as the contents are commonly called, out of the bucket (rather than putting water into the bucket).
- No more than a fixed number of tokens, B, can accumulate in the bucket, and if the bucket is empty, we must wait until more tokens arrive before we can send another packet.
- This algorithm is called the token bucket algorithm.

Prof. Prabhu Kichadi - 9880437187

Admission Control:

- It is a validation process in communication systems where a check is performed before a connection is established to see if current resources are sufficient for the proposed connection.
- The idea is simple: do not set up a new virtual circuit unless the network can carry the added traffic without becoming congested.
- One technique that is widely used to keep congestion that has already started from getting worse is admission control.
- Once congestion has been signaled, no more virtual circuits are set up until the problem has gone away.
- An alternative approach is to allow new virtual circuits but carefully route all new virtual circuits around problem areas. For example, consider the subnet of Fig. (a), in which two routers are congested, as indicated.

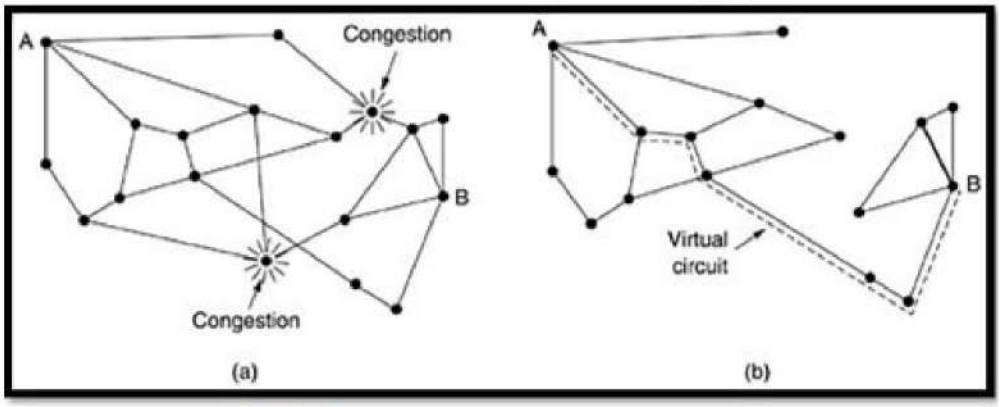


Figure 5-27. (a) A congested subnet. (b) A redrawn subnet that eliminates the congestion.

A virtual circuit from A to B is also shown.

(a) A congested subnet. (b) A redrawn subnet that eliminates the congestion. A virtualcircuit from A to B is also shown.

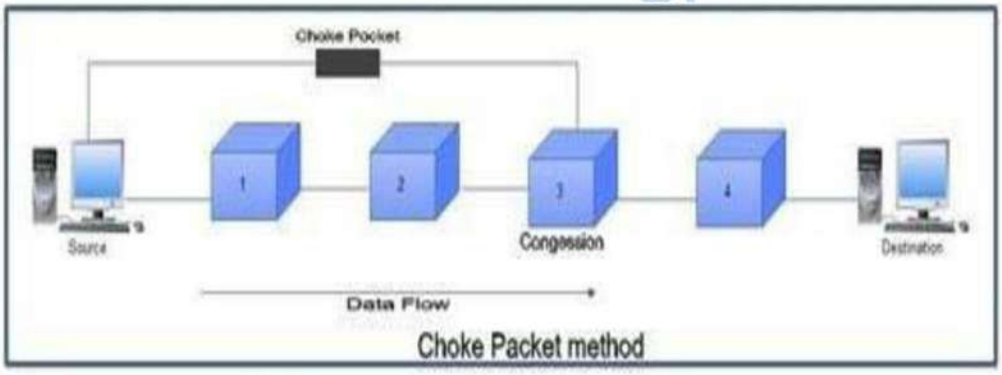
- Suppose that a host attached to router A wants to set up a connection to a hostattached to router B.
- Normally, this connection would pass through one of the congested routers.
- To avoid this situation, we can redraw the subnet as shown in Fig. (b), omittingthe congested routers and all of their lines.

- The dashed line shows a possible route for the virtual circuit that avoids thecongested routers.

Hop By Hop Choke Packets:

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.
- 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.



- In this approach, the router sends a choke packet back to the source host, giving it the destination found in the packet.
- The original packet is tagged (a header bit is turned on) so that it will not generate any more choke packets farther along the path and is then forwarded in the usual way.
- When the source host gets the choke packet, it is required to reduce the traffic sent to the specified destination by X percent. Since other packets aimed at the same destination are probably already under way and will generate yet more choke packets, the host should ignore choke packets referring to that destination for a fixed time interval. After that period has expired, the host listens for more choke packets

for another interval. If one arrives, the line is still congested, so the host reduces the flow still more and begins ignoring choke packets again. If no choke packets arrive during the listening period, the host may increase the flow again.

- The feedback implicit in this protocol can help prevent congestion yet not throttle any flow unless trouble occurs.

Prof. Prabhu Kichadi - 9880437187

Reference Link of YouTube	
Flooding	https://youtu.be/jheCH044aE8
Distance Vector Routing Algorithm	https://youtu.be/8APsvGgH_Og
Hierarchical Routing Algorithm	https://youtu.be/eo4yLlB2p6Q
Link State Routing Algorithm	https://youtu.be/XlVERy8oL68
Congestion Control	https://youtu.be/od6s_NMsApU
Leakey Bucket Algorithm	https://youtu.be/hrB5gml8AmY
Token Bucket Algorithm	https://youtu.be/XYBrO6ppVmk