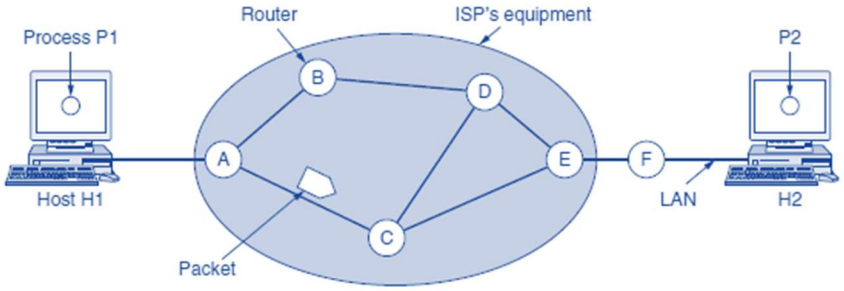
UNIT – III NETWORK LAYER:

1. **Design issues**
    - 1.1. Store and Forward Packet Switching
    - 1.2. Services provided to Transport Layer
    - 1.3. Implementation of Connectionless and connection –oriented Services
    - 1.4. Comparison of Virtual –Circuits and Datagram Networks
2. **Routing Algorithms**:
    - 2.1. Shortest Path Routing
    - 2.2. Flooding
    - 2.3. Hierarchical Routing
    - 2.4. Broadcast Routing,
    - 2.5. Multicast Routing,
    - 2.6. Distance Vector Routing
3. **Congestion Control** Algorithms
4. **Quality of Service**
5. **Internetworking**
6. **The Network layer in the internet.**

1. DESIGN ISSUES

1.1. Store and Forward Packet Switching:

- Let us understand how the Network Layer operate.
- The major components of networks are ISP's equipment i.e., Routers connected to the transmission line and then the customer equipment (Host H1 and Host H2) as shown in the figure below.



- Routers A to E forms the ISP equipment. Router F is outside ISP and not a part of ISP.
- Host H1(Home Computer) is directly connected to Router A in the ISP, consider it to be plugged to DSL modem.
- Host H2(office Computer) is a LAN through an Ethernet with router F and is on a leased line with the ISP equipment.
- But, the router F on the customer line considered to be part of the ISP's equipment as all the routers run on the same algorithm.
- The operation is as follows:
  - ✓ A Host intended to transmit the packet, sends the packet to its nearest Router
  - ✓ This packet is stored there until it is fully arrived at the router.
  - ✓ The link then processes the packet and verified it by generating the Checksum.
  - ✓ Then, the packet if forwarded to the next router until it reached the destination router.
  - ✓ This mechanism is called as ***Store-and-Forward Packet Switching***

1.2. Services provided to Transport Layer

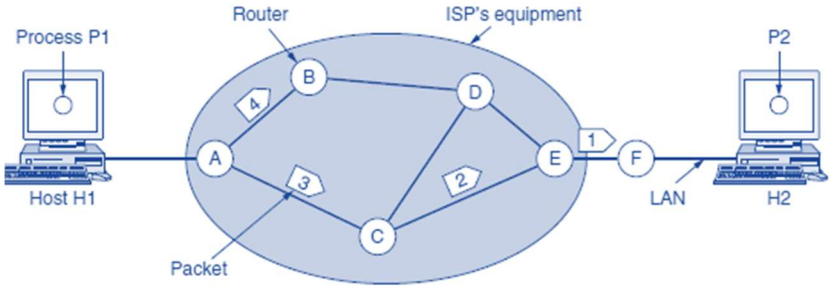
- A network layer provides services to the Transport Layer using the interface.
- These services to the transport layer are to be provided keeping in mind the following goals.
  - ✓ The services should be independent of the router technology
  - ✓ The transport layer should be shielded from the number, type, and topology of the routers present
  - ✓ The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs

1.3. Implementation of Connectionless and connection –oriented Services

- We already learnt that there are two types of services. i.e., Connectionless and Connection-Oriented Services.
- In connectionless service, the packets are injected individually and routed independently. i.e., no prior setup is required. Here, the packets are called as ***datagrams and the network is called as datagram network***
- In connection-oriented service, a connection has to be established all the way from source router to the destination router before a packet is being sent. This connection is called as ***Virtual Circuit*** and the network is called as ***Virtual Circuit Network***

Datagram Network(Connectionless):

- Consider P1 from Host H1 wants to send a message to P2 on Host H2 as shown below.
- It handovers the message to the Transport layer with an instruction to deliver the message to P2
- Host H1 runs the transport layer code
- It prepends a header to the front of the message, then hands-over to network layer.



- Let us now understand how the message is sent from Host H1 to Host H2
- Consider that the message is 4 times the packet size. Message now has to be split to 4 packet P1, P2, P3, P4
- Each packet is sent in-turns to Router A.
- The packet now enters ISP and ISP takes over the operations.
- Packets can be sent only on the directly connected lines. Router A has to two outgoing lines and the packets should be sent over these lines.
- Every router has its own internal table that tells where to send the packets to reach destination.
- Let the initial table of A is

A	-
B	B
C	C
D	B
E	C
F	C

And let, Packets P1, P2, P3 arrive at A. after the evaluation of checksum, these packets are forwarded to the routers based on the A's table. P1 is sent to C and then forwarded to E and F.

After reaching F (outside ISP), it is sent over LAN to reach P2 on Host H2. Also, P2 and P3 follows the same route.

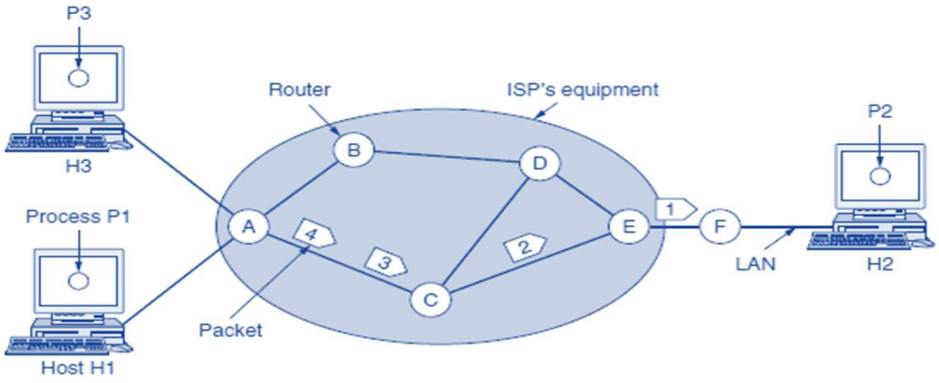
When P4 reached router A, it might have sensed a traffic-congestion in the previous path
(A-C-E) and P4 is sent over different route and it updates the table as below

A	-
B	B
C	C
D	B
E	B
F	B

- The algorithm that manages the tables and makes the routing decisions is called the **ROUTING ALGORITHM**

Virtual Circuit Network (Connection Oriented Service):

- The idea behind virtual circuits is to avoid having to choose a new route for every packet sent.
- when a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers.
- That route is used for all traffic flowing over the connection
- When the connection is released, the virtual circuit is also terminated.
- With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.
- Let us understand with an example shown below.



Here, Host *H1* has established connection-1 with Host *H2*. This connection is remembered as the first entry in each of the routing tables. The A's table be

A's Table					C's Table					E's Table			
IN		OUT			IN		OUT			IN		OUT	
H1	1	C	1		A	1	E	1		C	1	F	1
H3	1	C	2		A	2	E	2		C	2	F	2

The first line of *A*'s table says that if a packet bearing connection identifier 1 comes in from *H1*, it is to be sent to router *C* and given connection identifier 1. Similarly, the first entry at *C* routes the packet to *E*, also with connection identifier 1.

Now, let H3 also wants to establish the connection with H2. It chooses connection identifier 1 (because it is initiating the connection and this is its only connection) and tells the network to establish the virtual circuit. This leads to the second row in the tables.

Note that we have a conflict here because although *A* can easily distinguish connection 1 packets from *H1* from connection 1 packets from *H3*, *C* cannot do this. For this reason, *A* assigns a different connection identifier to the outgoing traffic for the second connection.

Avoiding conflicts of this kind is why routers need the ability to replace connection identifiers in outgoing packets.

This process is called as ***LABEL SWITCHING.***

2. ROUTING ALGORITHMS

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

Every packets need to be passed through multiple hops/nodes to reach the destination.

If the source and destination nodes are on different networks, the data transmission has to be handled carefully.

Routing Algorithm, is a method used in computer networks to determine the best path for data packets to travel from a source to a destination across a network. A Router is responsible for routing of data packets.

If the network uses datagram internally, the decision on the best route changes frequently.

If the network uses virtual circuit internally, the decision on the best route is made only when the virtual circuit is set-up. i.e., the data packets need to follow the already established route. (This is called as Session Routing as the route remains the same for the entire session.

A ***Router*** does two processes inside it,

1. Handles the packets that arrive
2. Decide on the outgoing path i.e., creation of routing table.

A ***Routing Algorithm*** is the procedure a router uses to determine the best path for data packets to travel from a source to a destination in a computer network. *The routing algorithm should be able to cope with changes in the topology*.

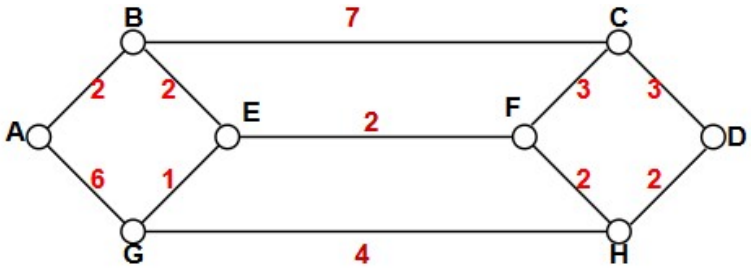
Routing algorithms can be grouped into two major classes.

1. **Non-Adaptive algorithm** (***Static Routing***):
   - ✓ Routes are manually set
   - ✓ Automatic updating of route isn't available; they are to be updated manually
   - ✓ Routing is hard to manage in larger networks
2. **Adaptive algorithm** (***Dynamic routing***):
   - ✓ Routes are updated automatically.
   - ✓ Because of which, the CPU usage is more.
   - ✓ Follows protocols to the route the packets.
   - ✓ Easily adapts to congestion during transmission.

Here, we will be discussing about various Routing Algorithms like

2.1. Shortest Path Routing (Dijkstra's Routing Algorithm)
2.2. Flooding
2.3. Hierarchical Routing
2.4. Broadcast Routing
2.5. Multicast Routing
2.6. Distance Vector Routing

- The main idea of the routing algorithm is to build a graph of the network, with each node of the graph representing a router and each edge of the graph representing a communication line, or link.
- In doing so, the algorithm chooses the shortest path between routers/nodes/hops.
- One way to measure the shortest path is to measure the path length by considering the **hops/nodes.** Other way is by finding the shortest path by **distance**. Considering the below network, ABE and ABC are equal if hops are considered, but using the distance ABC is greater than ABE



- The shortest path is the fastest path rather than the path with the fewest edges or kilometers.
- Several algorithms for computing the shortest path between two nodes of a graph are known. This one is due to Dijkstra (1959) and finds the shortest paths between a source and all destinations in the network

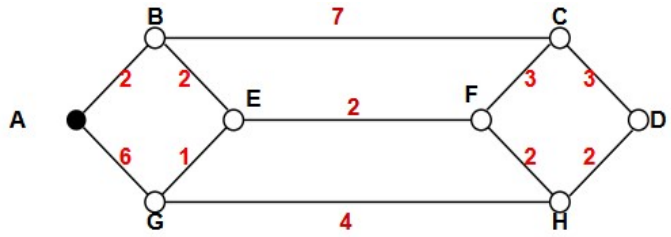
Dijkstra's Shortest Path Algorithm:

Algorithm is as follows:

- Each node is labelled along with the distance from source along the best shortest path.
- Distance must be non-negative.
- Initial conditions: since the path is unknown, all nodes are labelled as infinity.
- Labels are tentative initially. As the algorithm proceeds and the shorted path is found, the label becomes permanent.
- As the algorithm proceeds, nodes may change each time the shortest distance is found

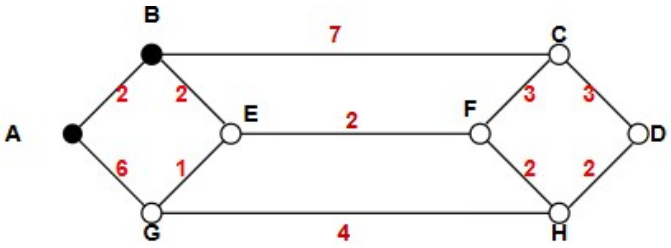
Let us understand the algorithm with an example below where the weights represent distance between nodes. Let us evaluate the shortest path between nodes A and D

Initially, Node A as marked permanent. The shortest distance to other nodes is not known and hence distance is marked as ∞ and destination is unknown. It is as shown below



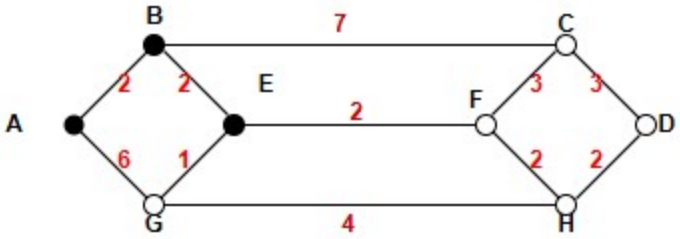
Node	Distance	Previous Node
A	0	A
B	∞	-
C	∞	-
D	∞	-
E	∞	-
F	∞	-
G	∞	-
H	∞	-

Next, we look into the adjacent cells of A i.e., **B** and **G** and label each with the distance from **A to B/G**. The node with the shortest path is labelled as permanent. In this case, node **B** becomes permanent as below.



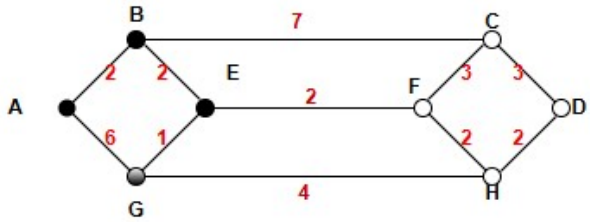
Node	Distance	Previous Node
A	0	A
B	**2**	**A**
C	∞	-
D	∞	-
E	∞	-
F	∞	-
G	**6**	**A**
H	∞	-

Next, we look into the adjacent cells of B i.e., **C** and **E** and label each with the distance from **A to B to C/E**. The label with the shortest path is labelled as permanent. In this case node **E** becomes permanent. It is as shown below.



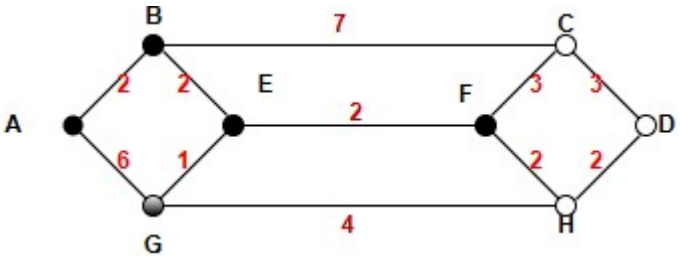
Node	Distance	Previous Node
A	0	A
B	2	A
C	**9**	**B**
D	∞	-
E	**4**	**B**
F	∞	-
G	6	A
H	∞	-

Now, the adjacent cells to E are **G** and **F.** Label G and F with the distances from **A to B to E to G/F.** It is observed that the shortest distance to G is now **5** and to F is **6.** of these two nodes , the shorted distance is to node G. Hence, G is made permanent node



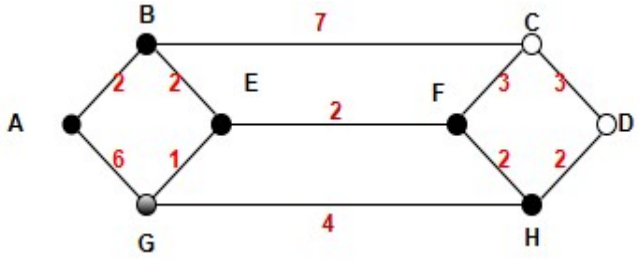
Node	Distance	Previous Node
A	0	A
B	2	A
C	9	B
D	∞	-
E	4	B
F	**6**	**E**
G	**5**	**E**
H	∞	-

Now, the adjacent node to G is H and the shortest distance is **A to B to E to G to H** with distance of 9. Of the remaining nodes F has the shortest distance and hence F is made permanent.



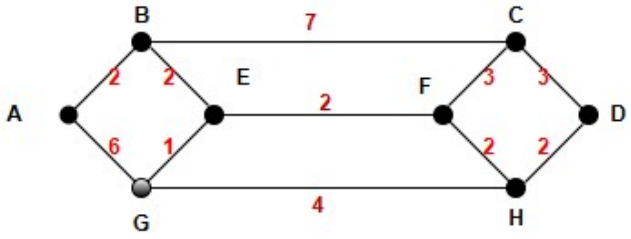
Node	Distance	Previous Node
A	0	A
B	2	A
C	9	B
D	∞	-
E	4	B
F	**6**	**E**
G	5	E
H	**9**	**G**

The adjacent node to F is C and H. The shortest distance is **A to B to E to F to C/H.**
The distance is 9 in either of the routes **A to B to C and A to B to E to F to C,** in this situation we retain the route with least number of nodes i.e., **A to B to C.**
Of the nodes C and H, H has the shortest distance and hence H is made permanent node.



Node	Distance	Previous Node
A	0	A
B	2	A
C	**9**	**B**
D	∞	-
E	4	B
F	**6**	**E**
G	5	E
H	**8**	**F**

The adjacent cell to H is D and hence the shortest route is **A to B to E to f to H t**



Node	Distance	Previous Node
A	0	A
B	2	A
C	9	B
D	**10**	**H**
E	4	B
F	6	E
G	5	E
H	8	F

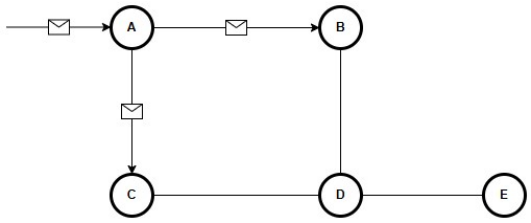
The shortest path is finally evaluated as **A to B to E to F to H to D** which is **10** as shown below

Node	Distance	Previous Node
A	0	A
B	~~2~~	A
C	9	B
D	~~10~~	H
E	4	B
F	~~6~~	E
G	5	E
H	~~8~~	F

2.2. Flooding:

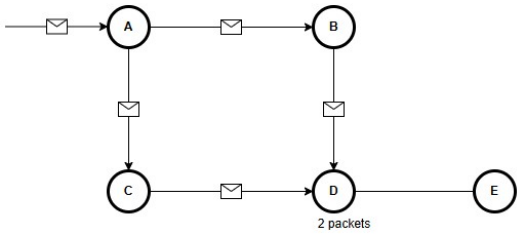
- When a routing algorithm is implemented; every node should take the decision based on the present situation. Flooding is a simple technique where every incoming packet is sent onto every outgoing like except the one on which it arrived.
- Flooding generated many duplicate packets (Multiple copies of the same packet travel through different paths). Some measures have to be taken to eliminate duplicate packets.
- One such measure to include the hop counter contained in the header of each packet. The hop count is decremented on each hop and finally the packet is discarded when the hop count reached zero.
- Consider the example below to understand Flooding In computer networks

Node A receives a packet and forwards the same on to its neighbouring nodes **B** and **C**

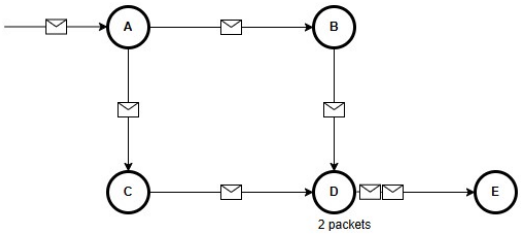


Node B receives the packet, forwards to its neighboring node **D** but not **A** (Data will not be sent back to the node from where it arrived).

Node C receives the packet, forwards to its neighboring node **D** only. This process creates flooding at node D.



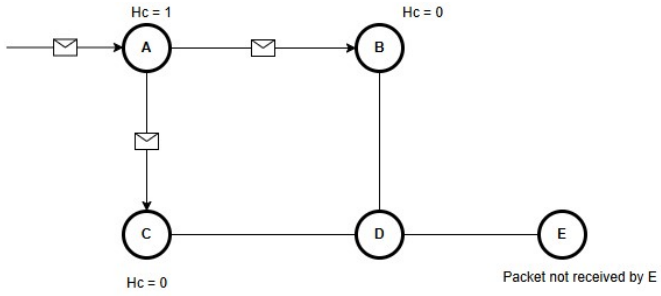
Node D now has 2 packets with same destination address of E, and forwards to its neighboring node **E** only. E now has 2 packets with same destination address.



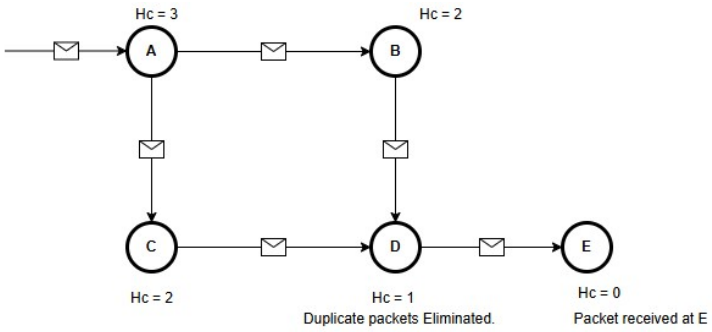
Sufficient measure has to be taken to eliminate duplicate packets.

To understand flooding,

Consider the hop-count(H_c) at node A be 1. When the packet is sent to B and C, the hop-count becomes 0 and the packet is discarded. Thus, the packet is received only by B and C but not the final node E



Consider another example with hop-count(H_c) at node A be 3. At B &C, the hop-count is 2, as the packet is forwarded to next node, the hop-count at D is 1 which eliminates flooding and finally at E it is 0. This indicated that the packet traversed in 2 different routes A→B→D→E and A→C→D→E, only one packet is received by E as shown below



2.3 Hierarchical Routing:

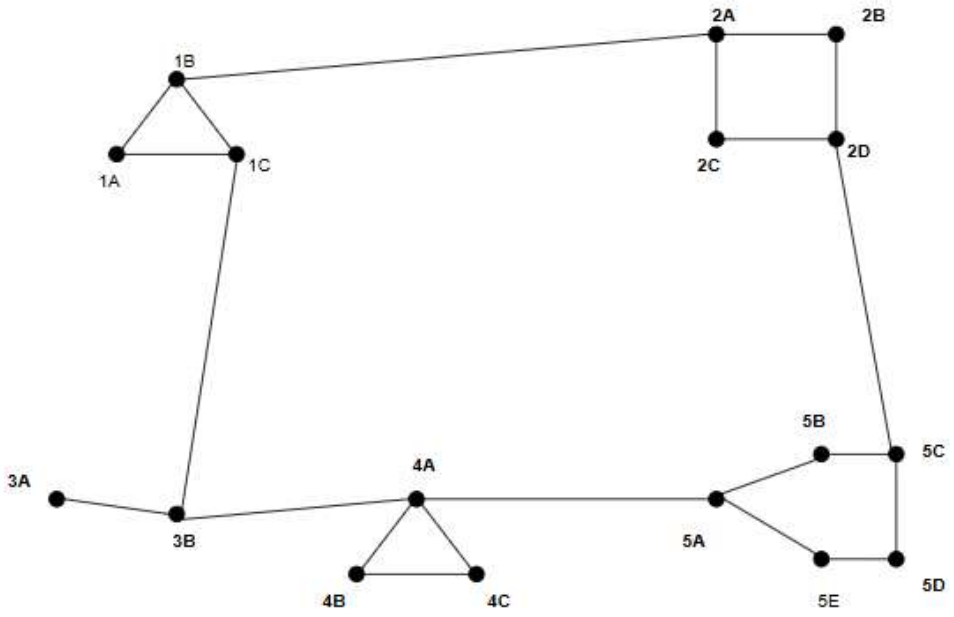
As the network grows, each node's routing table grows.

In this routing, the nodes are divided into **regions.** Each nodes knows all the details about how to route the data with the nodes within a region, but knows nothing about the internal structure of another region. This arrangement is termed as two-level hierarchy where we have nodes and regions.

As the network becomes huge, this two-level hierarchy is not sufficient. Hence we group region to form a cluster, group clusters to form a zone and so on.

To understand Hierarchical Routing, consider the network below where nodes

1A,1B,1C	form	region1
2A,2B,2C,2D	form	region2
3A,3B	form	region3
4A,4B,4C	form	region4
5A,5B,5C,5D,5E	form	region5



The nodes that connect the regions are called as Hub-Nodes. The data from one region to another will be router through the hub-nodes only.

Looking into the network, traffic from **region 1 to region 2** goes through the link **1B-2A** and traffic from **region1 to region3** goes through the link **1C-3B.**

In order to send data from node 1A of region 1 to nodes of other regions, we need to create table with 3 entries for region 1, 4 entries for region 2, 2 entries for region 3, 3 entries for region 4 and 5 entries for region 5. This creates the table for 1A with 17 entries. This can be reduced to 7 entries using Hierarchical Table.

The table consists of Destination, line (Hub-Node of Region 1) and no of hops to each the destination as shown below

Destination	Line (Hub-Node)	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

Full Table For 1A

Destination	Line (Hub-Node)	Hops
1A		
1B	1B	1
1C	1C	1
Region 2	1B	2
Region 3	1C	2
Region 4	1C	3
Region 5	1C	4

Hierarchical Table for 1A

Note: Data can be transferred to 5C via region2 as well as region 3. No. of hops via region 2 is 6 and via region3 is 5. Hence, data is transferred via the link consisting of less hops i.e. **1B** but not 1C

*For practice, prepare the Hierarchical table for data transferred from **2A to rest of regions***

2.4. BROADCAST ROUTING

In some applications, hosts need to send messages to many or all other hosts. Sending a packet to all destinations simultaneously is called **BROADCASTING.**

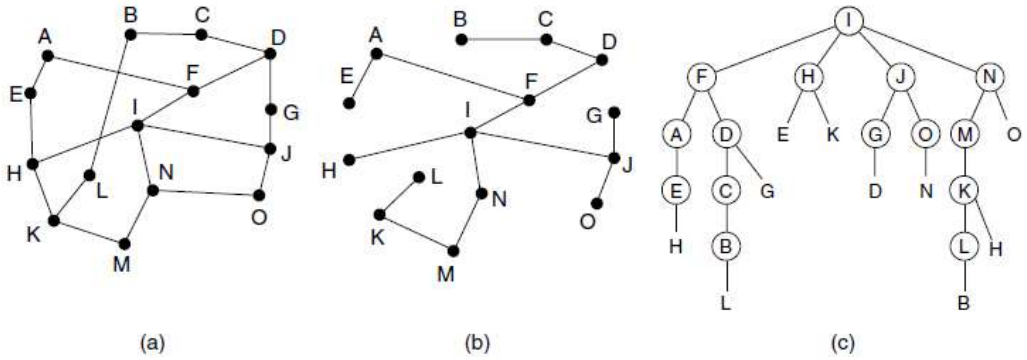
One broadcasting method is to send a distinct packet to each destination. This type of broadcasting utilizes more bandwidth, and the source requires addresses of all the destination, because of which this method not suggestible.

upgradation to the above method is **Multi-destination routing**

- each packet contains a list of destinations.
- When a packet arrives at a router, the router checks all the destinations 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 and includes in each packet only those destinations that are to use the line
- After a sufficient number of hops, each packet will carry only one destination like a normal packet
- The main drawback once again is that the source required to know all the destinations.

Upgradation to the above method is **Reverse Path Forwarding:**

- When a broadcast packet arrives at a router, the router checks to see if the packet arrived on the link that is normally used for sending packets *toward* the source of the broadcast.
- If the broadcast packet itself followed the best route from the router, the router forwards copies of it onto all links except the one it arrived on.
- If, however, the broadcast packet arrived on a link other than the preferred one for reaching the source, the packet is discarded as a likely duplicate.
- Consider an example with router I

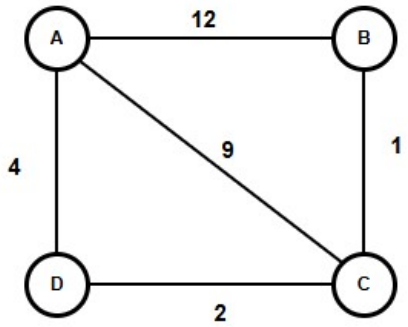


- Let the sink = I,
- on the first hop, I send packets to F, H, J, N. Each of these arrive on the preferred path to I and is indicated by circle.
- On the second hop, eight packets are generated, two by each of the routers that received a packet on the first hop. All eight of these arrive at previously unvisited routers, and five of these arrive along the preferred line. These 5 routers are rounded.

- On the third hop, six packets are generated and only E, C, K arrive on preferred path and the others (G, D, N) are duplicates.
- On the fourth hop, 4 packets are generated and only B and L arrive on preferred path and the rest are duplicates.
- On the fifth hop, 2 packets are generated and they do not arrive on preferred path. These are considered as duplicated.
- Broadcasting now terminates after 5 hops and 24 packets being generated
- Major advantage of is **Reverse Path Forwarding** is that it sends the broadcast packet over each link only once in each direction

2.6. Distance Vector Routing

- Computer networks generally use dynamic routing algorithms that are more complex than flooding, but more efficient because they find shortest paths for the current topology.
- A **distance vector routing** algorithm operates by having each router maintain a table (i.e., a vector) giving the best known distance to each destination and which link to use to get there.
- The distance vector routing algorithm is sometimes called by other names, most commonly the distributed **Bellman-Ford** routing algorithm
- In distance vector routing, each router maintains a routing table indexed by, and containing one entry for each router in the network.
- This entry has two parts:
  - **i.** the preferred outgoing line to use for that destination and
  - **ii.** an estimate of the distance to that destination.
- Consider the example below where the distance vector table is found in 2 stages only
  Note: For **n-nodes**, the shortest path using distance vector table can be calculated in **(n-1)** stages



Stage -1

Dest.	Dist.	Next Node
A	0	A
B	12	B
C	9	C
D	4	D

Vector Table - A

Dest.	Dist.	Next Node
A	12	A
B	0	B
C	1	C
D	∞	-

Vector Table - B

Dest.	Dist.	Next Node
A	9	A
B	1	B
C	0	C
D	2	D

Vector Table - C

Dest.	Dist.	Next Node
A	4	A
B	∞	-
C	2	C
D	0	D

Vector Table - D

Stage-2

Dest.	Dist.	Next Node
A	0	A
B	7	D, C
C	6	C
D	4	D

Vector Table - A

Dest.	Dist.	Next Node
A	7	A
B	0	B
C	1	C
D	3	C, A

Vector Table - B

Dest.	Dist.	Next Node
A	6	D
B	1	B
C	0	C
D	2	D

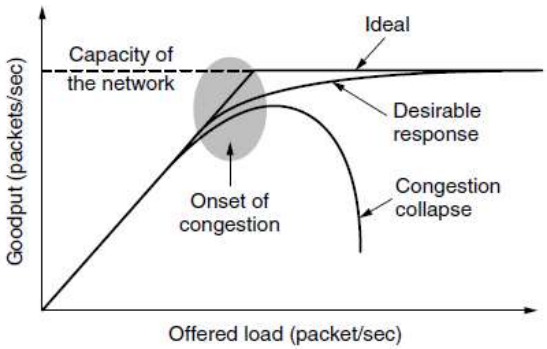
Vector Table - C

Dest.	Dist.	Next Node
A	4	A
B	3	C
C	2	C
D	0	D

Vector Table - D

3. Congestion Control Algorithms

- Congestion is the situation where too many packets present in the network causes either a packet delay or loss and degrades the performance.
- Since the congestion occurs in the network layer, it the responsibility of the network layer to reduce the congestion. Also, the transport layer has equal responsibility to reduce the congestion as it transports the packets.
- In other words, the transport layer and the network layer has to work together to reduce the congestion.
- Let us understand the concept of congestion with a graph below, drawn between **Goodput (packets/sec)** and **Offered load (packets/sec)**



Goodput (packets/sec) is the rate at which only the useful packets are delivered after dropping duplicates, retransmitted packets
Offered load (packets/sec) is the rate at which packets are sent into the network

The ***linear line*** indicates that the network can handle the traffic effectively with a zero scope of congestion. i.e., offered load matches the goodput.

When the ***offered load increases***, i.e., as it approaches the capacity of the network, long queues are formed at the nodes which delays the packets and a possible packet drop occurs. Under this condition, it is desired to ensure the load to be within the capacity of the network. This can be achieved by a proper congestion control mechanism. With this mechanism, the curve deviates from linearity but is well within the capacity of the network.

Without a proper congestion control mechanism, heavy traffic is injected and possible packet drops, packet retransmission and long queues occur which results in **congestion collapse**.

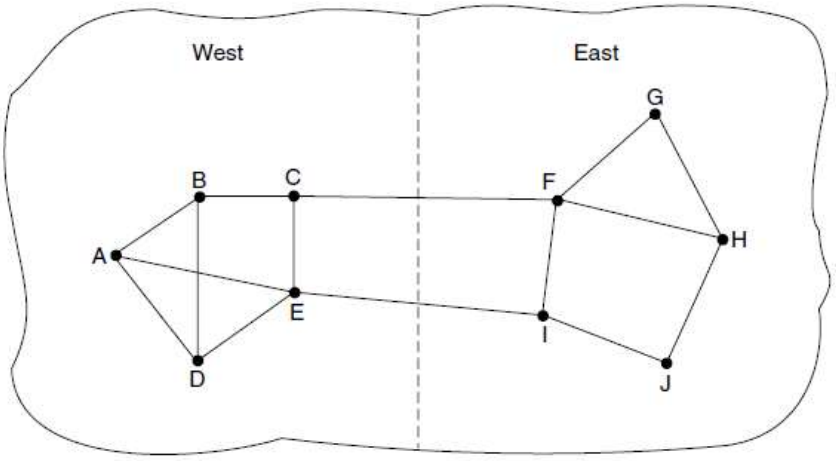
Note: Congestion mostly occurs in a low-bandwidth link.

3.1. Approaches to Congestion Control:

- The presence of congestion means that the load is greater than the network capacity.
- We can either increase the network capacity or decrease the load.
- Sometimes resources like backup router are added, or increase the bandwidth, or routers which are heavily loaded are upgraded. This is called as **Network Provision**.
- Network routes can be adjusted to match daily traffic shifts as users in different time zones go online and offline, making better use of existing capacity. This is called **Traffic-Aware Routing.**
- If increasing the capacity of the network is not possible, then we have to decrease the load to avoid congestion. Additional link may be added to beat the congestion in the network. This is called **Admission control.**
- If the congestion is imminent, then the network can request back the sources to **throttle** the traffic, or can **slow down** the traffic itself.
- Two difficulties with this approach arise. The first one is how to identify the start of congestion which can avoided by monitoring the average load and packet loss. The second one is how to inform the source that needs to slow down, this can be avoided by maintaining a proper time scale i.e., after every N packet the router will inform to STOP transmitting and after every internal of T(nanoseconds) the router will inform to SEND.
- If all the above mechanisms fail, the network is forced to shed the packets that it cannot deliver. This is called as **Load Shedding.**

Traffic Aware Routing:

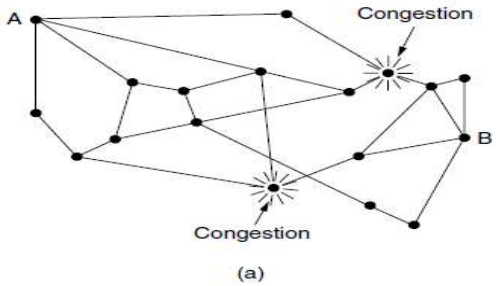
- The routing schemes we learnt till now can adopt to change in topology, but nor for the changes in load.
- One way is to set the link weight to be a function of the (fixed) link bandwidth and propagation delay plus the (variable) measured load or average queuing delay
- Traffic-aware routing can be understood using the example below.



- Consider the network above, which is divided into two parts, East and West, connected by two links, *CF* and *EI*. Suppose that most of the traffic between East and West is using link *CF*, and, as a result, this link is heavily loaded with long delays. Including queueing delay in the weight used for the shortest path calculation will make *EI* more attractive. After the new routing tables have been installed, most of the East-West traffic will now go over *EI*, loading this link. Consequently, in the next update, *CF* will appear to be the shortest path. As a result, the routing tables may oscillate wildly, leading to erratic routing and many potential problems.
- If load is ignored and only bandwidth and propagation delay are considered, this problem does not occur. Attempts to include load but change weights within a narrow range only slow down routing oscillations.
- Two techniques can contribute to a successful solution. The *first* is multipath routing, in which there can be multiple paths from a source to a destination. In our example this means that the traffic can be spread across both of the East to West links. The *second* one is for the routing scheme to shift traffic across routes slowly enough that it is able to converge, as in the scheme of Gallagher (1977). Given these difficulties, in the Internet routing protocols do not generally adjust their routes depending on the load. Instead, adjustments are made outside the routing protocol by slowly changing its inputs. This is called **traffic engineering**.

3.2. Admission Control:

- The basic idea of congestion control using Admission control is "Not to set up a new virtual circuit unless the network can carry the added traffic without becoming congested ".
  Ex: In the telephone system, when a switch gets overloaded it practices admission control by not giving dial tones
- Admission control can also be combined with traffic-aware routing by considering routes around traffic hotspots as part of the setup procedure
- consider the network illustrated



Fig(a) Congested Network

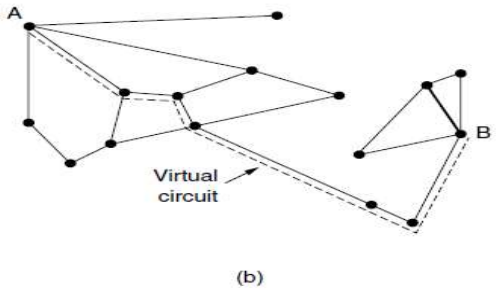


Fig (b) : The portion of the network that is not congested.

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