INTRODUCTION:

The network layer is the lowest layer that deals with end-to-end transmission. That means source system to destination system data transfer by providing the possible best routing by using the router and hops.

For this, the network layer must know about the entire topology of the network which includes the set of all routers and links. By suing those network layer can choose appropriate path.

All the protocols in the network layer use **"store-and forward packet switching**" approach. In this, when a sender forwards the packet, it is stored there until it has fully arrived and the link has finished its processing by verifying the checksum.

This process is called as **store-and forward packet switching** approach. This approach is applied between each and every router and hop, until the packet reaches the destination.

Each router contains routing table which are including the details of their performed data transfer, such as destination, link, packet sequence number…etc.

The network layer also use the "**Routing algorithms**" to make the decision about path and to maintain the routing tables at each router

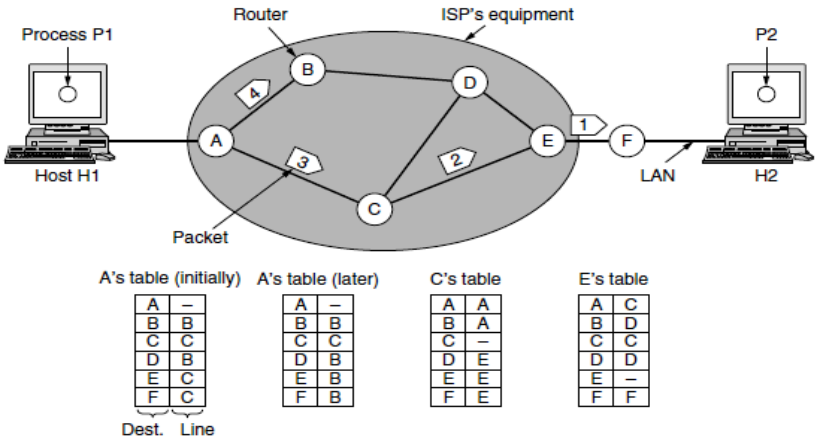
The network layer can offer different services to its upper layer (Transport layer). From those we are having two major services.

1) Connection-oriented network service. (Virtual-circuit network).
2) Connection-less network service. (Datagram network).

DATAGRAM NETWORK:

It is generally called as "connection-less network". That means it does not contain any specific connection or link or path between source and destination by using routers. It is having only one phase that is "**data transfer phase**".
If connectionless service packets are send into the network individually and routed independently from each other. That means the packets are routed or forwarded in different directions by using different set of routers in the network.

Because of this reason the network is called **datagram network**. In this the packet is called as **datagram.**



In the above dig, the data is divided into 4 datagrams. Each datagram is forwarded from source to destination by using routers available. Here we are having two systems (host's) H1 and H2 and 6 routers A,B,C,D,E and F.

In the dig, the datagrams forwarded from source H1 to destination H2. For this the H1 forwards the datagrams 1,2 ,3 and 4 to router A . Router A forwards the datagrams 1,2,3 to router C followed by router E and F to reach the destination H2. But datagram 4 is routed in different direction.
The router A forwards the datagram 4 by using Router B followed by D,E and F to reach the destination.

Because of this it is clear that all of the datagrams are not forwarded in the same direction and they followed different paths to reach the destination.

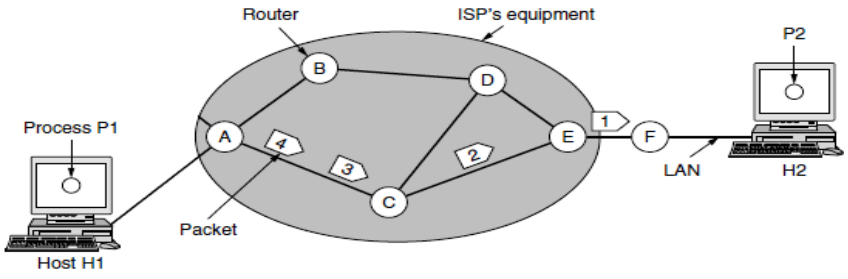
In datagram network:

1) There is **no** setup phase and teardown phase. It is only having data transfer phase.
2) There is no flow control and error control.
3) Datagrams contain sequence numbers. But no ACK.
4) At sender side all datagrams are forwarded in order , but at receiver side the datagrams are not received in order.
5) It is difficult to maintain reliability, and quality of service.
6) The router table only contains the destination address and the line to receive the next router or destination.
7) It does not provide and information about loss or failure of data.
8) It is difficult to maintain congestion control.
9) It does not provide retransmission of failed or loss data.

VIRTUAL-CIRCUIT NETWORK:

The idea behind virtual circuits is to avoid the selection of new route for every packet sent. This VC network is also called as "connection-oriented network".
That means it contains the 3 phase:

1) Connection Establishment phase.
2) Data transfer phase.
3) Tear down phase.



IN		OUT	
H1	1	C	1

A Routing table

IN		OUT	
A	1	E	1

C Routing table

In the above dig, the data is divided into 4 Packets. Each packet is forwarded from source to destination by using routers available. Here we are having two systems (host's) H1 and H2 and 6 routers A,B,C,D,E and F.

In the dig, the packets forwarded from source H1 to destination H2. For this the H1 forwards the packets 1,2 ,3 and 4 to router A . Router A forwards the packets to router C followed by router E and F to reach the destination H2. All the packets follow the same path to reach the destination , because it is connection oriented and its path is already established before the data transfer.

In VC network:

1) There is setup phase, teardown phase and data transfer phase.
2) There is flow control and error control.
3) Each Packet contains sequence number and ACK along with VCI.
4) At sender side all datagrams are forwarded in order , at receiver side the datagrams are also received in order.
5) It is easy to maintain reliability, and quality of service.
6) The router table contains the source and destination address and sequence number with IN and OUT details.
7) It provides and information about loss or failure of packet.
8) It is easy to maintain congestion control.
9) It can provide the time-out and retransmission.

ROUTING ALGORITHMS:

In the network layer the main responsibility is providing routing for the incoming data packets to reach its destination. For this purpose the "**Routing algorithms**" are used in network layer. Routing algorithms are the part of network layer and used to make the **decision about path** and to maintain the routing tables at each router
Routing algorithms can make variable path and permanent path for data transfer.

That means, if we are using **connection oriented network** (or) virtual-circuit network, these algorithms will establish a dedicated path between source and destination for data transfer. All packets must follow the established (same) dedicated path to reach the destination.

If we are using **connection less network** (or) datagram network, these algorithms does not establish a path between source and destination for data transfer. **Each packet will be treated as separate**, and **individually**, these algorithms will provide different paths to reach the destination based on the situations.

Routing algorithms can be grouped into two major classes:

1) **Non-Adaptive** (or) **static routing algorithms**.
2) **Adaptive** (or) **Dynamic routing algorithms**.

Non-Adaptive routing algorithms **do not change their path (or) route** for the failures (or) heavy-traffic in the path. First, static algorithms will establish a path and data transfer will follow only that route whatever happens.
Adaptive algorithms can change their routing decisions (or) path with reflect to changes in the network topology, changes in the traffic. These algorithms will get the information locally, from adjacent routers, or from all routers in the network when they change the routes.

To get the proper information about the network and the routers, these routing algorithms will use different types of metrics:

1) No.of hops (station or system).
2) Distance.
3) Transmission time.
4) Delay.

DIJKSTRASHORTEST PATH ROUTING ALGORITHM:

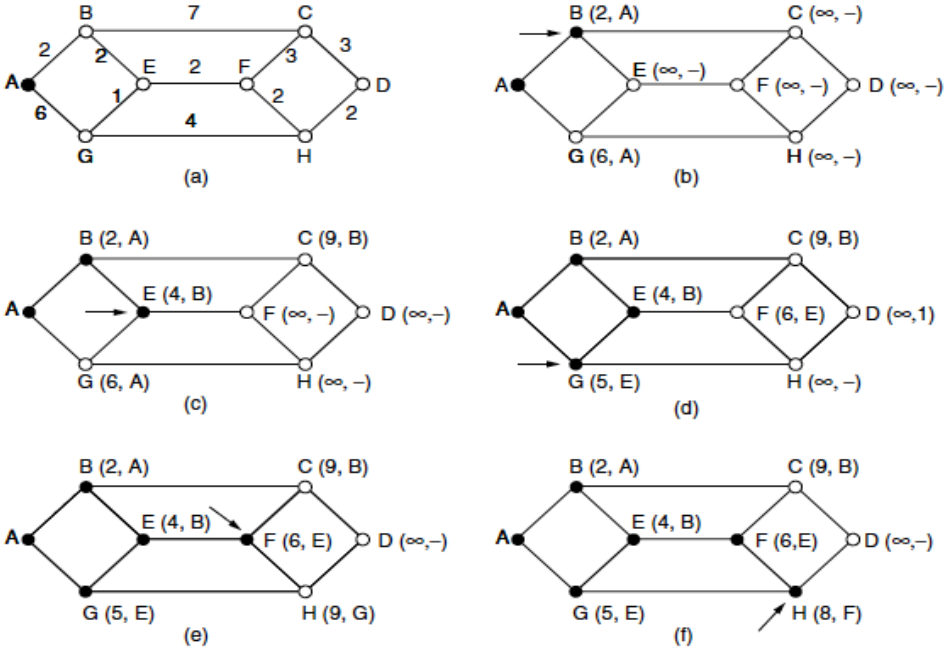
The basic responsibility of the network layer is to provide best and shortest path from source to destination by using available routers in the network. We are having no.of algorithms to find out the shortest path. But mostly used shortest path algorithms is "**Dijkstra's Shortest Path Routing Algorithm**".

In this algorithm we use "**distance**" as metric. That means we can identify distance between one router to other routers.

In this algorithm each node or router is having a label. The label contains two parts:

1) Total count of Distance.
2) Its source node.

Initially, all nodes are labeled with $(\infty, -)$, infinity indicates the distance and -- indicates the source node.



In the above given graph, all the nodes are **indicated with distance** from other node. We first identify the source and destination.

After that all nodes are labeled as $(\infty, -)$ .except the source node.

This algorithms process starts from source node, and by visiting the adjacent nodes and identifying the shortest distance and **replacing the label's** with total count distance and its source node, select the adjacent node with shortest distance and continue the process until algorithm reaches the destination node. The processing node is indicated by using and arrow. When the path is permanent **the node circle is filled**.

We start our process from source node A by visiting its **adjacent nodes B and G** , relabeling the values as B(2,A) , G(6,A) and identifying the shortest distance from **A is router B with (2,A).**

Now the process starts with B, by visiting its adjacent nodes C, E, relabeling the nodes as C(9,B), E(4,B). Identifying the shortest distance from B is router **E with (4,B).**

Now the process starts with E, by visiting its **adjacent nodes G, F**, relabeling the nodes as G(5,E), F(6,E). Identifying the shortest distance from E is router **G with (5,E).**

Here first G is visited as an adjacent node to A and labeled with **A** related values. But G is not processed. An un-processed node can relabeled or visited many No.of time as long as it is not processed.

Now the process starts with G, by visiting its **adjacent node H**, **A is source node** and already processed. So, we only re-label the node H as H(9,G). But it is long path comparing to F(6,E). So we **get back** and Identifying the **shortest distance from E** is **router F with (6,E).**

Now the process starts with F, by visiting its **adjacent nodes C, H**, relabeling the nodes as C(9,F), H(8,F). Identifying the shortest distance from F is router **H with (8,F).**

Now the process starts with H, by visiting its adjacent **node D** which is our destination, relabeling the nodes as **D(10,H).**

Now we are having the **permanent (or) fixed (or) dedicated path** from source A to destination D:
A → B(2,A) → E(4,B) → F(6,E) → H(8,F) → D(10,H)

FLOODING:

Flooding is a simple local technique, in this a router can forward every incoming packet to **every outgoing line** it is connected, except the line packet is arrived.

The concept of Flooding:

1) It generates many numbers of duplicate packets.
2) It is waste to implement when we are having only one single destination.
3) It is most useful to forward the packet to reach many No.of destinations.
4) It is still used in sending the radio signals send to all available devices in the range.
5) It can provide many No.of ways to reach the destination.
6) It can find out the shortest path effectively.
7) It can find out a new path to reach the destination even a path is failed.
8) It is the best concept than many algorithms to provide best minimum delay.
9) But the bandwidth is wasted because of duplicate packets.
10) The packet must contains a sequence number, to identify either the packet is duplicated or first coming to the system.

DISTANCE VECTOR ROUTING ALGORITHM:

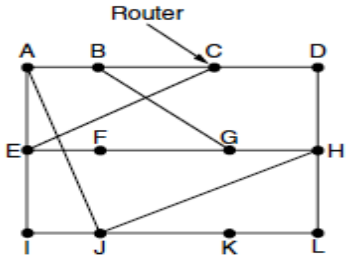
It is one of the Dynamic routing algorithms. Sometimes it is also called as the distributed **Bellman-Ford** routing algorithm.

That means in each **router is having a table**, and in that routing table, we are having the details about **other router** and **distance to reach that router** from this router, and a **possible link or direction** to reach that router.

These routing tables are updated by exchanging information with the neighbors. That means, this algorithm finds a path based on the **information given by other router**.

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: the preferred **outgoing line** to use to reach that destination and an estimate of the **distance** to that destination.



From above subnet we are having routers named as A,B,C,D,E,F,G,H,I,J,K and L. In this algorithm each router is having some **directly connected** neighbors. It means each router will know very well about its neighbors. Anyone router wants to reach other router (not its neighbor) it must take detail or information from it directly connected neighbors.

Consider the following diagrams subnet. Select the router "**J**". Router J is having **4 neighbor routers** A, H, I and K.
Router J is having (here metric is delay) **delay** to reach its neighbors:
Delay From
J-to- A =8,
J-to- I =10,
J-to- H =12,
J-to- K =6.

In this algorithm each neighbor contains the "delay" information in its routing table for all other routers in the network.
That means, the neighbors A, I, H, K is having routing table **with delay details** to reach all other routers from it.
In the above dig the neighbors A, I, H, K are maintaining routing tables with delay to all other routers.
Now the selected router J will **update its own routing table** with delay to all other routers in the network.
For this updating, the router J will **trust the delay information** from its neighbors A, I ,H K.

Because J is having only direct link with A, I, H, K, but it does not have any direct link with other routers, to reach other routers in the network J must use the delay details from its neighbors and **one shortest path as link from its neighbors**.

If J wants to reach any other router (not its neighbor) it computes the delay by using:
J - To – Neighbor + Neighbor - To – Other Router for each of its neighbor.

After processing all of its neighbors, J will select the shortest delay from all results and updates it routing table with that **short delay and neighbor** to follow the link.

A table

To	delay
A	0
B	2
C	6
D	8
E	10
F	13
G	14
H	10
I	36
J	8
K	30

I table

To	Delay
A	36
B	19
C	13
D	15
E	7
F	10
G	11
H	3
I	0
J	10
K	19

H table

To	Delay
A	10
B	8
C	3
D	11
E	14
F	20
G	5
H	0
I	3
J	2
K	21

K table

To	delay
A	30
B	12
C	32
D	6
E	22
F	21
G	22
H	21
I	19
J	32
K	0

Now the router J will compute the shortest delay to other routers by using its neighbors delay details and updates its NEW routing table.

J New Routing table

To	Short delay	line
A	**8**	**A**
B	10	A
C	14	A
D	12	K
E	17	I
F	20	I
G	17	H
H	**12**	**H**
I	**10**	**I**
J	0	------
K	**6**	**K**

To reach from J to B :

1) J to A + A to B = 8 + 2 = 10.
2) J to I + I to B = 10 + 19 = 29.
3) J to H + H to B = 12 + 8 = 20.
4) J to K + K to B = 6 + 12 = 18.

So, after computations the shortest delay from J to B is 10 by using line A. it is updated in J table.

To reach from J to C :

1) J to A + A to C = 8 + 6 = 14.
2) J to I + I to C = 10 + 13 = 23.
3) J to H + H to C = 12 + 3 = 15.
4) J to K + K to C = 6 + 32 = 38.

So, after computations the shortest delay from J to C is 14 by using line A. it is updated in J table.

To reach from J to D :

1) J to A + A to D = 8 + 8 = 16.
2) J to I + I to D = 10 + 15 = 25.
3) J to H + H to D = 12 + 11= 23.
4) J to K + K to D = 6 + 6 = 12.

So, after computations the shortest delay from J to D is 12 by using line K. it is updated in J table.

To reach from J to E :

1) J to A + A to E = 8 + 10 = 18.
2) J to I + I to E = 10 + 7= 17.
3) J to H + H to E = 12 +14 = 26.
4) J to K + K to E = 6 + 22= 28.

So, after computations the shortest delay from J to E is 17 by using line I. it is updated in J table.

To reach from J to F :

1) J to A + A to F = 8 + 13 = 21.
2) J to I + I to F = 10 + 10 = 20.
3) J to H + H to F = 12 +20 = 32.
4) J to K + K to F = 6 + 21 = 27.

So, after computations the shortest delay from J to F is 20 by using line I. it is updated in J table.

To reach from J to G :

1) J to A + A to G = 8 + 14 = 22.
2) J to I + I to G = 10 + 11 = 21.
3) J to H + H to G = 12 + 5= 17.
4) J to K + K to G = 6 + 22 = 28.

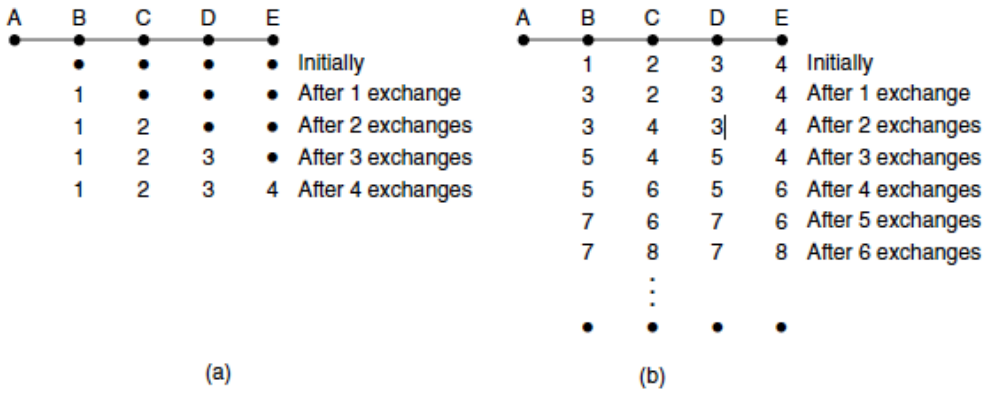
So, after computations the shortest delay from J to G is 17 by using line H. it is updated in J table.

In the distance vector routing algorithm, this process is applied to each and every router when it wants to update its new routing table. Here the neighbor routers information is the most important. It must be correct and accurate. Otherwise it will leads to biggest problems called “count – to - infinity ”.

THE COUNT-TO-INFINITY PROBLEM:

Distance vector routing is useful, but it has a serious drawback in practice.
We know that this routing algorithms is completely depends on the neighbors information to update its routing table.

As long as the neighbors provide the correct information there will be no problem. If any neighbor provides the **wrong information** when updating a routing table, it leads to a serious problem called "**Count-to-Infinity Problem**".



Consider above diagram, 5 routers are connected in serial, and we are using metric "No.of hops".
That means **No.of stations** are having **in between** source to destination.

The diagram contains the No.of hops distance from router A to other router in serial order.
When router A is not available initially all routers are indicating 0.
When router A is available in the first exchange router B updates the value as 1. Router B is having only one station (no.of hops=1) distance from A.
After seeing router B update, in the next exchange router C updates its value to 2. After seeing router C update, in the next exchange router D updates its value to 3. After seeing router D update, in the next exchange router E updates its value to 4.

Now let us consider the situation in which all the links and routers are initially up. Routers *B*, *C*, *D*, and *E* have distances to *A* of 1, 2, 3, and 4 hops, respectively.

Router B is forwarding packet to router A and all of sudden if router A goes down, router B does not hear any reply form the router A. in this situation router B does not know anything to do.

In this time, router C is coming with a statement that "I am having link to router A" with distance 2.
Router B thinks that C is having multiple links to multiple destinations; from those links one of the link is reaching the router A.

By considering information ("wrong information from neighbor") from router C, in the first exchange router B updates its value to 3.

B-to-C is =1 and C-to-A =2 so totally B-to-C-to-A=3.

By seeing the router B value in the next exchange router C immediately updates its value to 4.

Still router B is unable to get router A. so it again verifies Router C value.

After seeing router C update, in the next exchange router B again updates its value to 5 (because c value is 4). Also router D updates its value to 5.

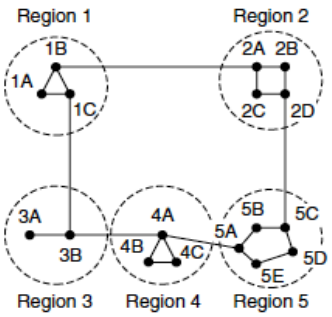
After seeing router B update, in the next exchange router C updates its value to 6.
After seeing router D update, in the next exchange router E updates its value to 6.

Still router B is unable to get router A. so it again verifies Router C value. After seeing router C update, in the next exchange router B again updates its value to 7 (because c value is 6). Also router D updates its value to 7.
After seeing router B update, in the next exchange router C updates its value to 8.
After seeing router D update, in the next exchange router E updates its value to 8.

All things happened due to wrong information from the neighbor, so each router will lead that the count to infinity. So, this problem is called as "count-to-infinity" problem.

HIERARCHICAL ROUTING:

Generally from above routing algorithms, each router is having table and in that table it contains the details of all other routers in the network. If the no.of routers are limited there will be no problem.
If the **network increases** also there is an **increase in the no.of routers count** as well as the increase in the **entries** of table. It will lead to big problem; it takes much time to scan and more bandwidth to send status.
To solve this problem, in hierarchical routing all the routers are divided into no.of **regions**.
Each router **knows all the details** about how to route packets to destinations within its **own region** but **knows nothing** about the internal structure of **other regions**.



(a)

Full table for 1A

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

(b)

Hierarchical table for 1A

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

(c)

Form above dig, we are having 5 regions with local routers. In all regions each local router knows everything about other local router. In each region **one of the routers** is connecting to **other router** in the **other region**.

Consider from region 1: to reach region 2 we use 1B-2A line. To reach region 3 we use 1C-3B line. Similar to this we are having links from one region to other region.

The main goal of hierarchical routing is to reduce the burden of the routing table updates by reducing the no.of entries in the routing table.

In the above diagram we are a general routing table with 17entries that means each router is having all other router entries in the table with line and no.of hops.

The last table shows, the hierarchical algorithms reduces the entries in the routing table by only giving a possible entry to **all local routers** and only one entry to each region.

BROADCASTING:

Sending a packet to all destinations simultaneously is called **broadcasting**. In this concept the source simply send a distinct packet to each destination in the network.

This concept is Not only wasteful of bandwidth but also slow and it also requires the source to have a complete list of all destinations.

For performing the broadcasting in effective manner, various methods are proposed .

1) Flooding.
2) Multi-destination Routing.
3) Reverse path forwarding.
4) Spanning tree.
5) Multicast routing.

Multi-Destination Routing:

In multi-destination routing, in which each packet contains **a list of desired destinations**.
When a packet arrives at a router, the router checks all the destinations and 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.

Reverse Path Forwarding:

The **reverse path forwarding** is elegant and simple.
When the broadcast packet arrives at a router, the router checks if the broadcast packet arrived on the link that is normally used for sending packets and it leads *toward* the source system of the broadcast packet. It line is used by the source to send the packets , there is an excellent chance that the broadcast packet is accepted by router and forwards the board cast packet to all of its available lines as a new copy.

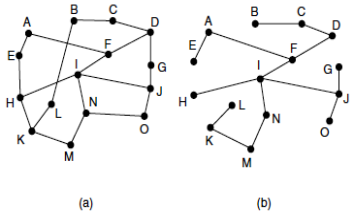
If the router identifies the board cast packet coming line is not regular towards the source, then the packet is considered as duplicate and it is discarded.

Spanning Tree :

A **spanning tree** is a subset of the network that includes **all the routers** but contains **no loops**.

Sink trees are spanning trees. If each router knows which of its lines belong to the spanning tree, it can copy an incoming broadcast packet onto all the spanning tree lines except the one it arrived on.

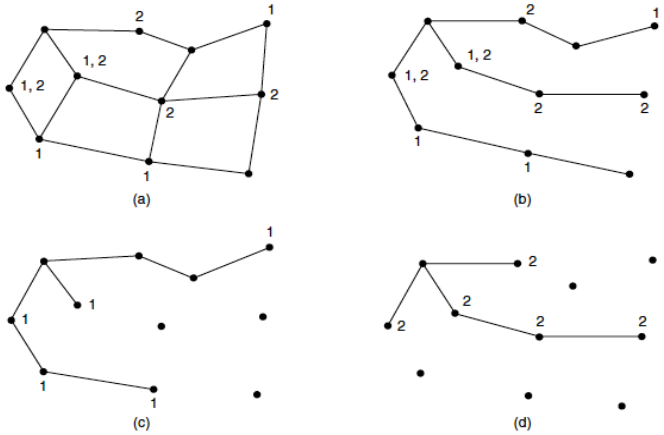
This method makes excellent use of bandwidth, generating the absolute **minimum number of packets** necessary to do the job.



Multicast Routing :

The concept of Sending a message or packet to a group is called **multicasting**. The routing algorithm used for this multicasting is called as **multicast routing**.

All multicasting schemes require some way to create and destroy groups and to identify which routers are members of which group.



In the above dig: we are having two groups 1and 2. Some router are belongs to both of the groups indicated with 1,2 on them. In this concept each incoming packet must contain group number. Like 1 or 2 or 1,2.

Based on the group identifier, the subnet will be prepared as a spanning tree.

That means, if the incoming board cast packet contains the group identifier as 1, in the subnet we only highlights those routers which are having group identifier 1 as a spanning tree and the packet is delivered to only those routers.

If the incoming board cast packet contains the group identifier as 2, in the subnet we only highlights those routers which are having group identifier 2 as a spanning tree and the packet is delivered to only those routers.

If the incoming board cast packet contains the group identifier as both (1,2) in the subnet we only highlights those routers which are having group identifier as both (1,2) as a spanning tree and the packet is delivered to only those routers.