III-UNIT

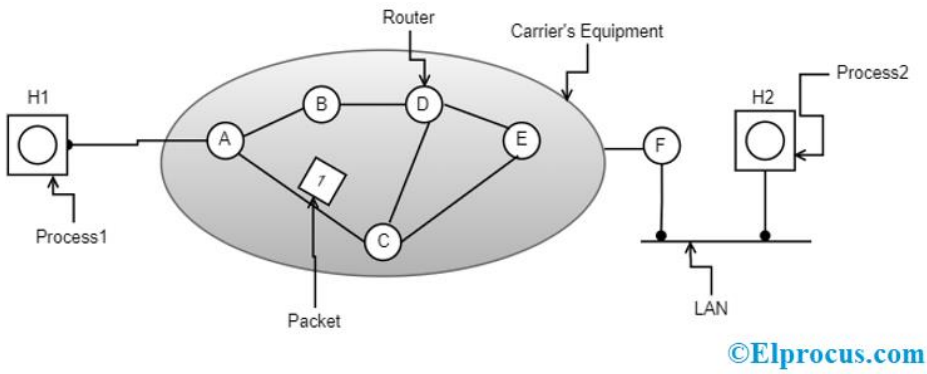
NETWORK LAYER

Network Layer Design Issues

Network layer comes up with certain design issues and they can be described as below:

1). Store-and-Forward Packet Switching

Here, the foremost elements are the carrier's equipment (the connection between routers through transmission lines) and the customer's equipment.



store-and-forward packet switching

- H1 has a direct connection with carrier router 'A', while H2 is connected to carrier router 'F' on a LAN connection.
- One of the carrier router 'F', is pointed outside the carrier's equipment as it does not come under the carrier, whereas considered as protocols, software, and construction.
- This switching network performs as Transmission of data happens when the host (H1) with a packet transfers it to the nearby router through LAN (or) point-to-point connection to the carrier. The carrier stores the packet until it completely arrives thus confirms the checksum.
- Then after, the packet is transmitted over the path until H2 is reached.

2). Services Provided to the Transport Layer

Through the network/transport layer interface, the network layer delivers its services to the transport layer. One might come across the question of what type of services does the network layer provides?

So, we shall move with the same query and find out the services offered.

Services offered by the network layer are outlined considering few objectives. Those are:

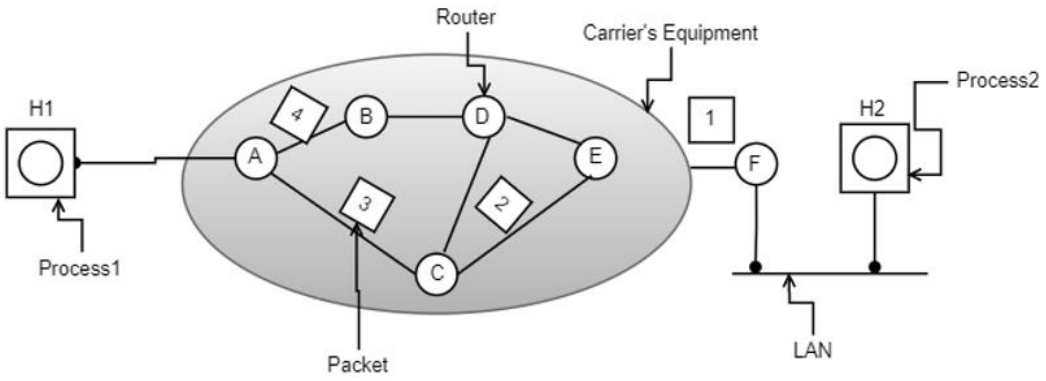
- Offering services must not depend on router technology
- The transport layer needs to be protected from type, number and the topology of the available routers.
- Network addressing the transport layer needs to follow a consistent numbering scenario also at LAN and WAN connections.

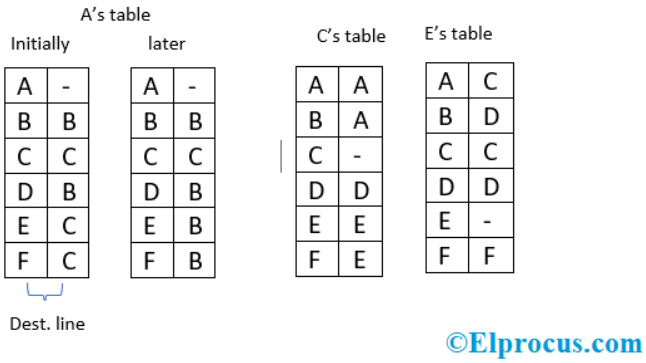
Note: Next comes the scenario of connection-Oriented or connectionless
Here, two groupings are possible based on the offered services.

Connectionless – Here, routing and insertion of packets into subnet is accomplished individually. No additional setup is necessary
Connection-Oriented – Subnet must offer reliable service and all the packets are transmitted over a single route.

3). Implementation of Connectionless Service

In this scenario, packets are termed as datagrams and the corresponding subnet is termed as datagram subnet. Routing in datagram subnet is as follows:





datagram subnet truth table

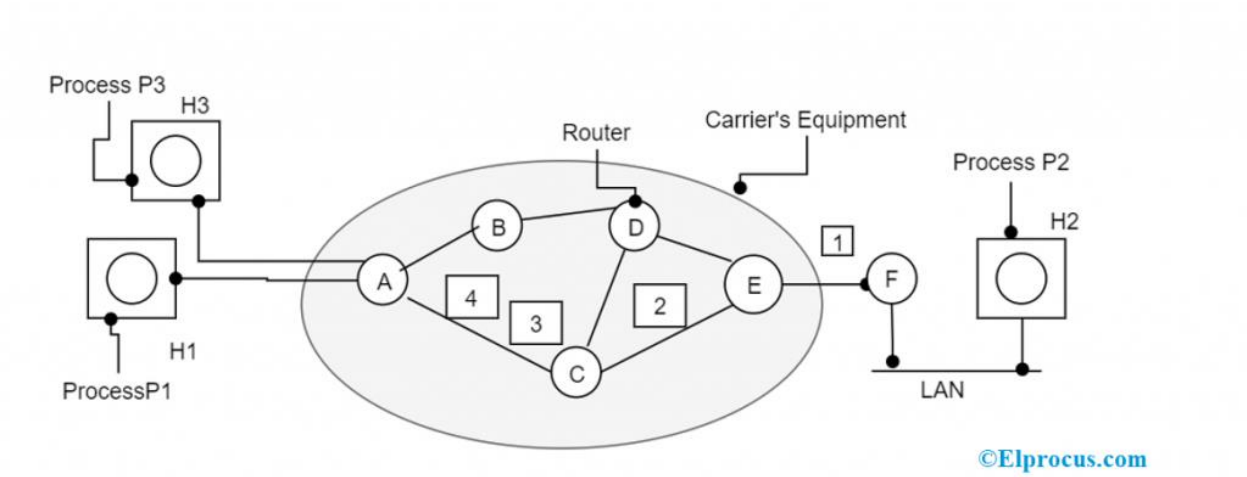
When the message size that has to be transmitted is 4 times the size of the packet, then the network layer divides into 4 packets and then transmits each packet to router 'A' through a few protocols. Each router is provided with a routing table where it decides the destination points.
In the above figure, it is clear that packets from 'A' need to be transmitted either to B or C even when the destination is 'F'. The routing table of 'A' is clearly outlined above.

Whereas in the case of packet 4, the packet from 'A' is routed to 'B', even the destination node is 'F'. Packet 'A' chooses to transmit packet 4 through a different path than the initial three paths. This might happen because of traffic congestion along the path ACE. So, the

4). Implementation of Connection-Oriented Service

Here, the functionality of connection-oriented service works on the virtual subnet. A virtual subnet performs the operation of avoiding a new path for each packet transmission. As a substitute for this, when there forms a connection, a route from a source node to a destination node is selected and maintained in tables. This route performs its action at the time of traffic congestion.

At the time when the connection is released, the virtual subnet also gets dismissed. In this service, every packet carries its own identifier that states the exact address of the virtual circuit. The below diagram shows the routing algorithm in the virtual subnet.



Implementation of Connection-Oriented Service

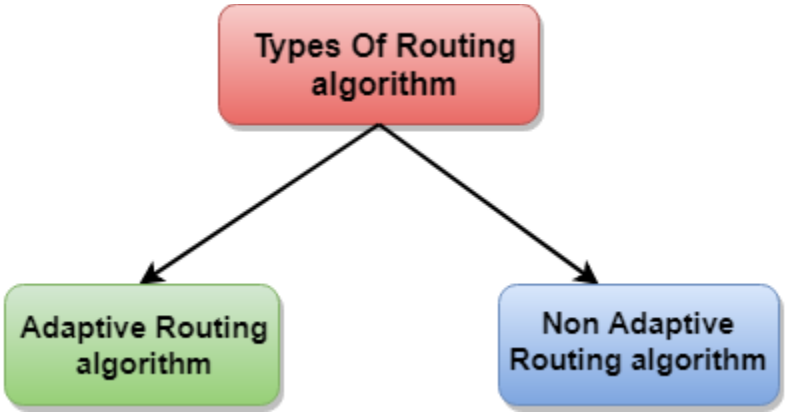
Routing algorithm

- 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
- Non-adaptive Routing algorithm



Adaptive 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.

An adaptive routing algorithm can be classified into three parts:

- **Centralized algorithm:** It is also known as global routing algorithm as it computes the least-cost path between source and destination by using complete and global knowledge about the network. This algorithm takes the connectivity between the nodes and link cost as input, and this information is obtained before actually performing any calculation. **Link state algorithm** is referred to as a centralized algorithm since it is aware of the cost of each link in the network.
- **Isolation algorithm:** It is an algorithm that obtains the routing information by using local information rather than gathering information from other nodes.
- **Distributed algorithm:** It is also known as decentralized algorithm as it computes the least-cost path between source and destination in an iterative and distributed manner. In the decentralized algorithm, no node has the knowledge about the cost of all the network links. In the beginning, a node contains the information only about its own directly attached links and through an iterative process of calculation computes the least-cost path to the destination. A Distance vector algorithm is a

decentralized algorithm as it never knows the complete path from source to the destination, instead it knows the direction through which the packet is to be forwarded along with the least cost path.

Non-Adaptive 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.

The Non-Adaptive Routing algorithm is of two types:

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.

Random walks: In case of random walks, a packet sent by the node to one of its neighbors randomly. An advantage of using random walks is that it uses the alternative routes very efficiently.

Differences b/w Adaptive and Non-Adaptive Routing Algorithm

Basis Of Comparison	Adaptive Routing algorithm	Non-Adaptive Routing algorithm
Define	Adaptive Routing algorithm is an algorithm that constructs the routing table based on the network conditions.	The Non-Adaptive Routing algorithm is an algorithm that constructs the static table to determine which node to send the packet.
Usage	Adaptive routing algorithm is used by dynamic routing.	The Non-Adaptive Routing algorithm is used by static routing.
Routing decision	Routing decisions are made based on topology and network traffic.	Routing decisions are the static tables.

Categorization	The types of adaptive routing algorithm, are Centralized, isolation and distributed algorithm.	The types of Non Adaptive routing algorithm are flooding and random walks.
Complexity	Adaptive Routing algorithms are more complex.	Non-Adaptive Routing algorithms are simple.

Distance Vector Routing Algorithm

- **The Distance vector algorithm is iterative, asynchronous and distributed.**
  - **Distributed:** It is distributed in that each node receives information from one or more of its directly attached neighbors, performs calculation and then distributes the result back to its neighbors.
  - **Iterative:** It is iterative in that its process continues until no more information is available to be exchanged between neighbors.
  - **Asynchronous:** It does not require that all of its nodes operate in the lock step with each other.
- The Distance vector algorithm is a dynamic algorithm.
- It is mainly used in ARPANET, and RIP.
- Each router maintains a distance table known as **Vector**.

Three Keys to understand the working of Distance Vector Routing Algorithm:

- **Knowledge about the whole network:** Each router shares its knowledge through the entire network. The Router sends its collected knowledge about the network to its neighbors.
- **Routing only to neighbors:** The router sends its knowledge about the network to only those routers which have direct links. The router sends whatever it has about the network through the ports. The information is received by the router and uses the information to update its own routing table.
- **Information sharing at regular intervals:** Within 30 seconds, the router sends the information to the neighboring routers.

Distance Vector Routing Algorithm

Let $d_x(y)$ be the cost of the least-cost path from node x to node y. The least costs are related by Bellman-Ford equation,

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

Where the minv is the equation taken for all x neighbors. After traveling from x to v, if we consider the least-cost path from v to y, the path cost will be $c(x,v)+d_v(y)$. The least cost from x to y is the minimum of $c(x,v)+d_v(y)$ taken over all neighbors.

With the Distance Vector Routing algorithm, the node x contains the following routing information:

- For each neighbor v, the cost c(x,v) is the path cost from x to directly attached neighbor, v.
- The distance vector x, i.e., $D_x = [D_x(y) : y \text{ in } N]$, containing its cost to all destinations, y, in N.
- The distance vector of each of its neighbors, i.e., $D_v = [D_v(y) : y \text{ in } N]$ for each neighbor v of x.

Distance vector routing is an asynchronous algorithm in which node x sends the copy of its distance vector to all its neighbors. When node x receives the new distance vector from one of its neighboring vector, v, it saves the distance vector of v and uses the Bellman-Ford equation to update its own distance vector. The equation is given below:

$d_x(y) = \min_v\{ c(x,v) + d_v(y)\}$ for each node y in N

The node x has updated its own distance vector table by using the above equation and sends its updated table to all its neighbors so that they can update their own distance vectors.

Algorithm

At each node x,

Initialization

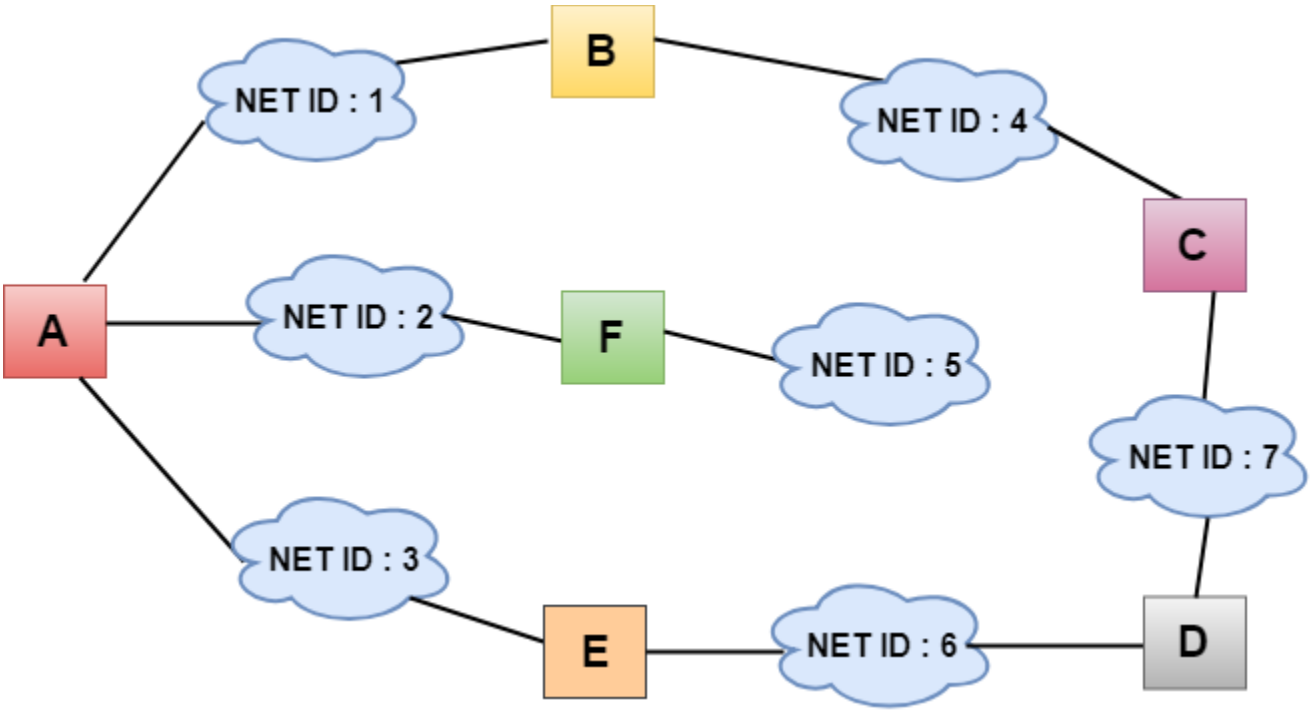
for all destinations y in N:
$D_x(y) = c(x,y)$ // If y is not a neighbor then $c(x,y) = \infty$
for each neighbor w
$D_w(y) = ?$ for all destination y in N.
for each neighbor w
send distance vector $D_x = [D_x(y) : y \text{ in } N]$ to w
loop
wait(until I receive any distance vector from some neighbor w)
for each y in N:
$D_x(y) = \text{minv}\{c(x,v)+D_v(y)\}$
If $D_x(y)$ is changed for any destination y
Send distance vector $D_x = [D_x(y) : y \text{ in } N]$ to all neighbors

forever

Note: In Distance vector algorithm, node x update its table when it either see any cost change in one directly linked nodes or receives any vector update from some neighbor.

Let's understand through an example:

Sharing Information

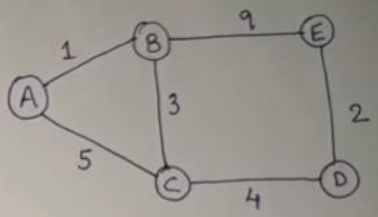


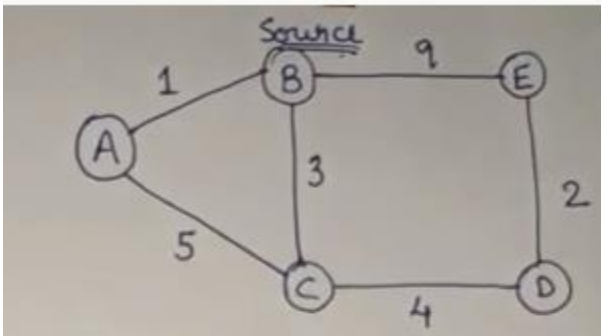
- In the above figure, each cloud represents the network, and the number inside the cloud represents the network ID.
- All the LANs are connected by routers, and they are represented in boxes labeled as A, B, C, D, E, F.
- Distance vector routing algorithm simplifies the routing process by assuming the cost of every link is one unit. Therefore, the efficiency of transmission can be measured by the number of links to reach the destination.
- In Distance vector routing, the cost is based on hop count.

DISTANCE VECTOR ROUTING

- It is iterative, distributed & asynchronous.
- It is dynamic algorithm.
- Each router maintains a distance table known as Vector.
- Let $d_x(y)$ be the least cost path from node x to node y. The least costs are related by Bellman Ford equation:

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





Cost = no. intermediate nodes should be minimum.

⇒ Routing table for hop 'B':

Destination	Cost	Next hop
A	1	A
C	3	C
E	9	E
D		

① B to A

B-A → 1 ✓
B-C-A → 8
B-E-D-C-A → 20

② B to C

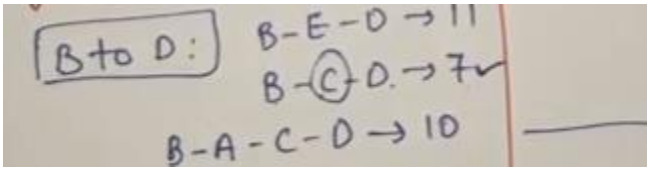
B-C → 3 ✓
B-E-D-C → 15
B-A-C → 6

③ B to E

✓ B-E → 9 ✓
✓ B-C-D-E
B-A-C-D-E → 1

B-C-D-E=9

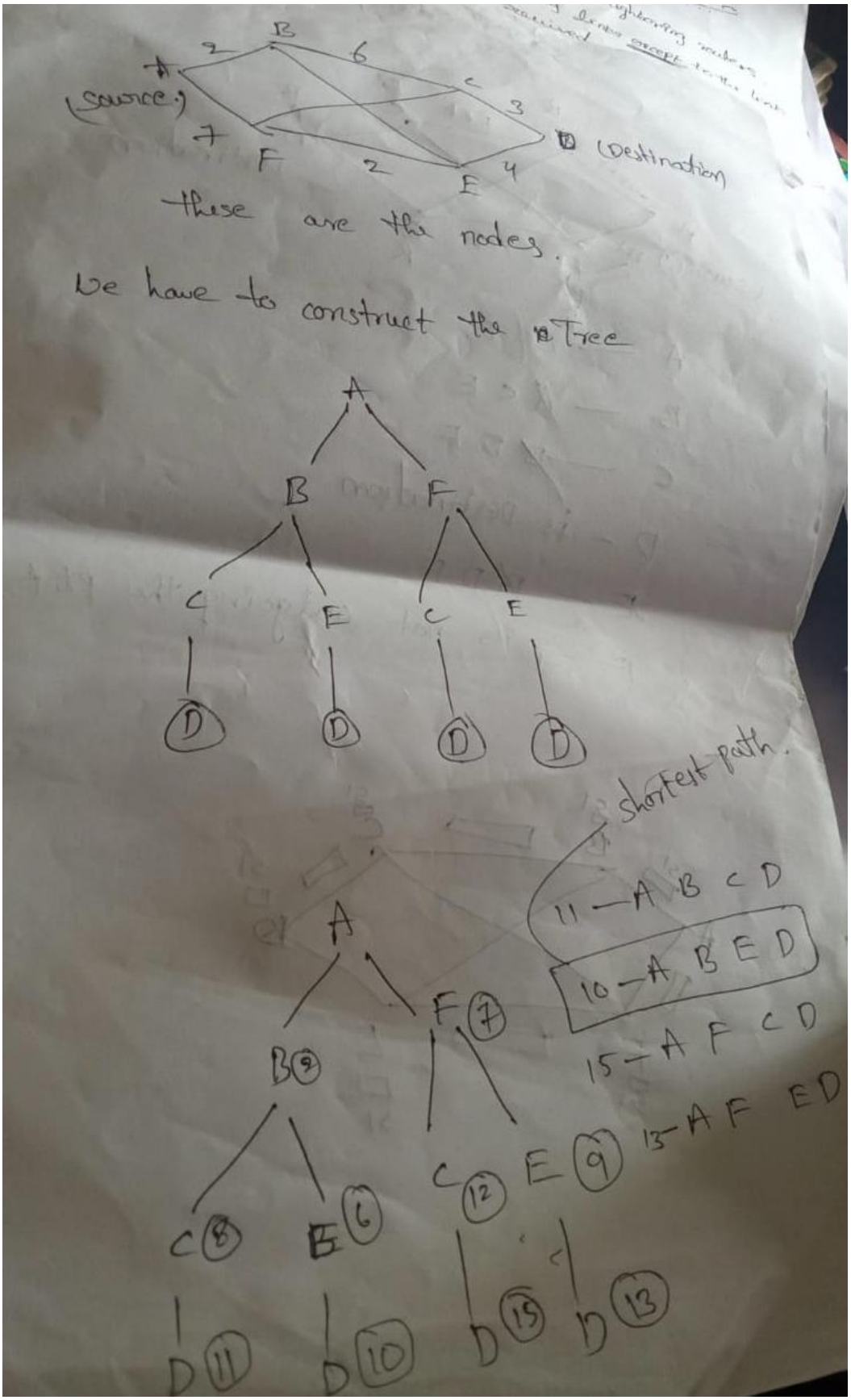
B-A-C-D-E=12



Shortest Path Routing

(a nonadaptive routing algorithm)

- Given a network topology and a set of weights describing the cost to send data across each link in the network
- Find the shortest path from a specified source to all other destinations in the network.
- Shortest path algorithm first developed by E. W. Dijkstra


B
2
6
A
C
3
(source)
7
F
2
E
4
D (Destination)
these are the nodes.
We have to construct the Tree
A
B
F
C
E
C
E
D
D
D
D
shortest path.
A
F (7)
B (2)
C (8)
B (6)
C (12)
E (9)
D (11)
D (10)
D (15)
D (13)
11 — A B C D
10 — A B E D
15 — A F C D
13 — A F E D

Example of Shortest Path Routing

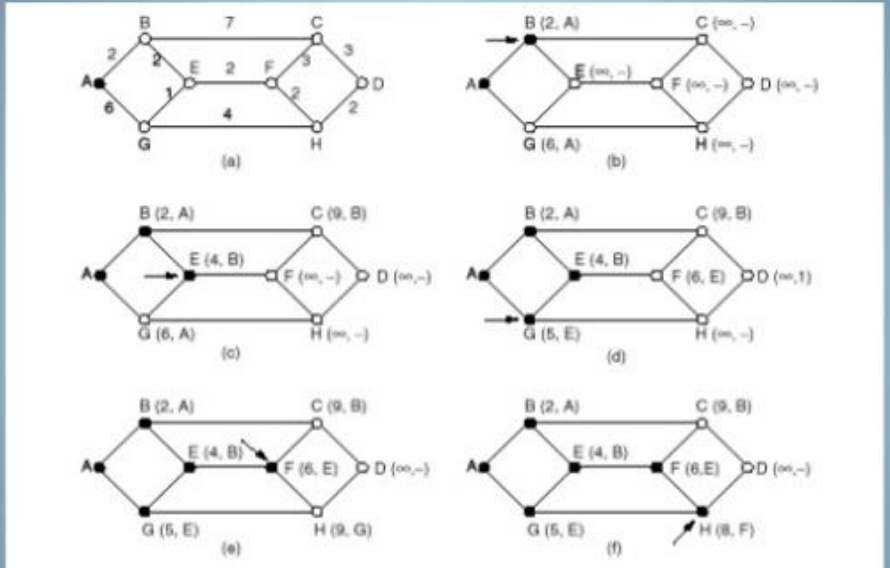


Fig. 5-6. The first five steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

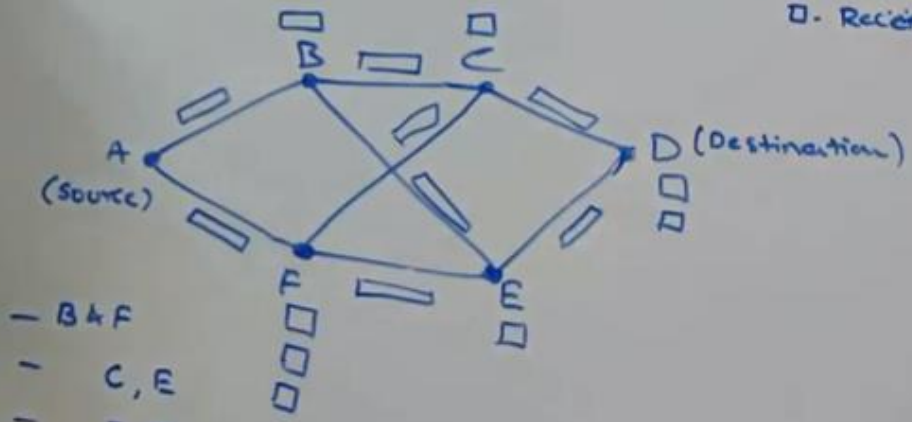
Flooding

(a nonadaptive routing algorithm)

- No network information is required
- Packet send by node to every neighbor
- Incoming packets retransmitted on every link without incoming link
- Eventually a numbers of copies will arrives at destination
- Each packet is uniquely numbered so duplicate can be discarded
- Nodes can remember packets already forwarded to keep network load in bounds

Flooding

* Broadcast the packet
* Sends the packet to all outgoing links except to the link from which it was received.



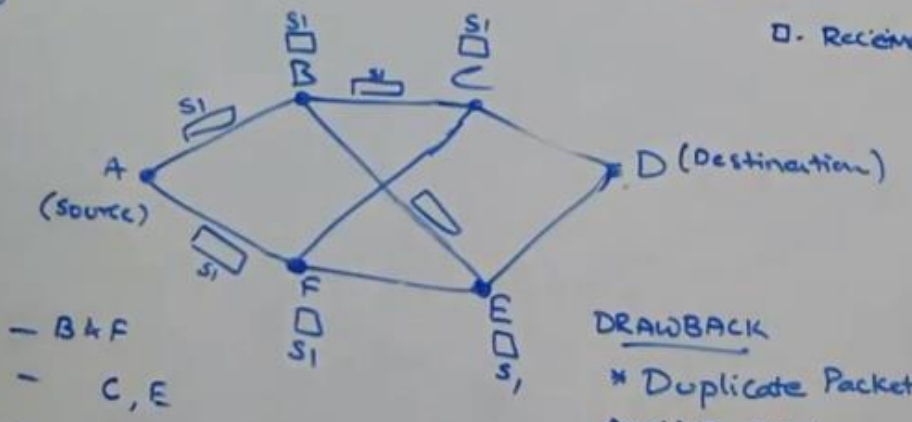
□ - Received

A – B & F
B – C, E
C – F, D
E – F, D

Flooding

* Broadcast the packet
* Sends the packet to all outgoing links except to the link from which it was received.

SEQUENCE NUMBER



□ - Received

A – B & F
B – C, E
C – F, D
E – F, D

DRAWBACK
* Duplicate Packets

Avoid Duplicates
* Use SEQUENCE NUMB

Flooding

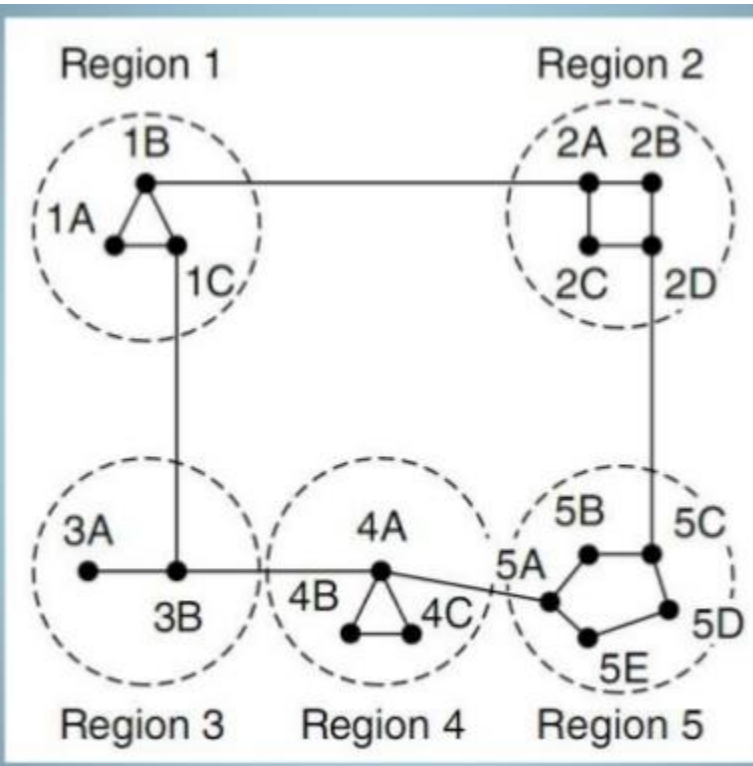
(a nonadaptive routing algorithm)

- All nodes are visited
- All possible routes are tried

- Selective Flooding
  - Flood only in the direction of the destination
- Practical example
  - Mobile when user 1 dial number to call user 2,the mobile station/company searches for user 2 in whole country if user 2 is out of reach. Then this process take few seconds
  - Distributed Databases

Hierarchical Routing

- Addresses the growth of routing tables
- Routers are divided into regions
- Routers know the routes for their own regions only
- Works like telephone routing
- Possible hierarchy
  - city, state, country, continent

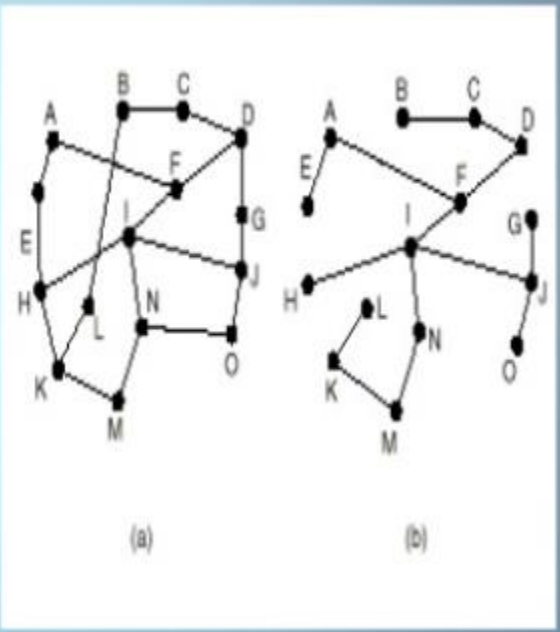


Broadcast Routing

- Send a separate packet to each destination
- Use flooding
- Use multidestination routing
  - Each packet contains a list of destinations
  - Routers duplicate packet for all matching outgoing lines
- Use spanning tree routing
  - a subset of the subnet that includes all routers but contains no loops.

Spanning Tree Broadcasting

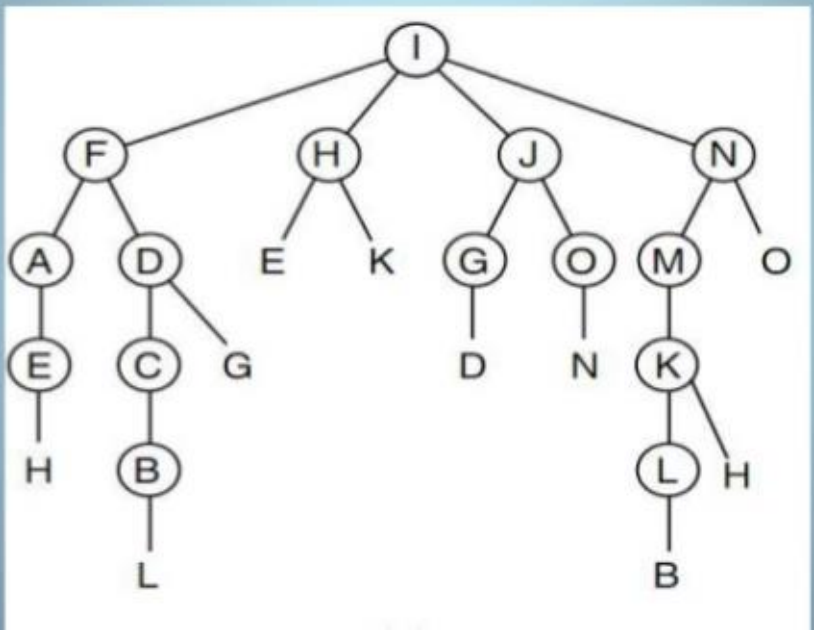
- Uses the minimum number of packets necessary
- Routers must be able to compute spanning tree
  - Available with link state routing
  - Not available with distance vector routing



Broadcast Routing (continued)

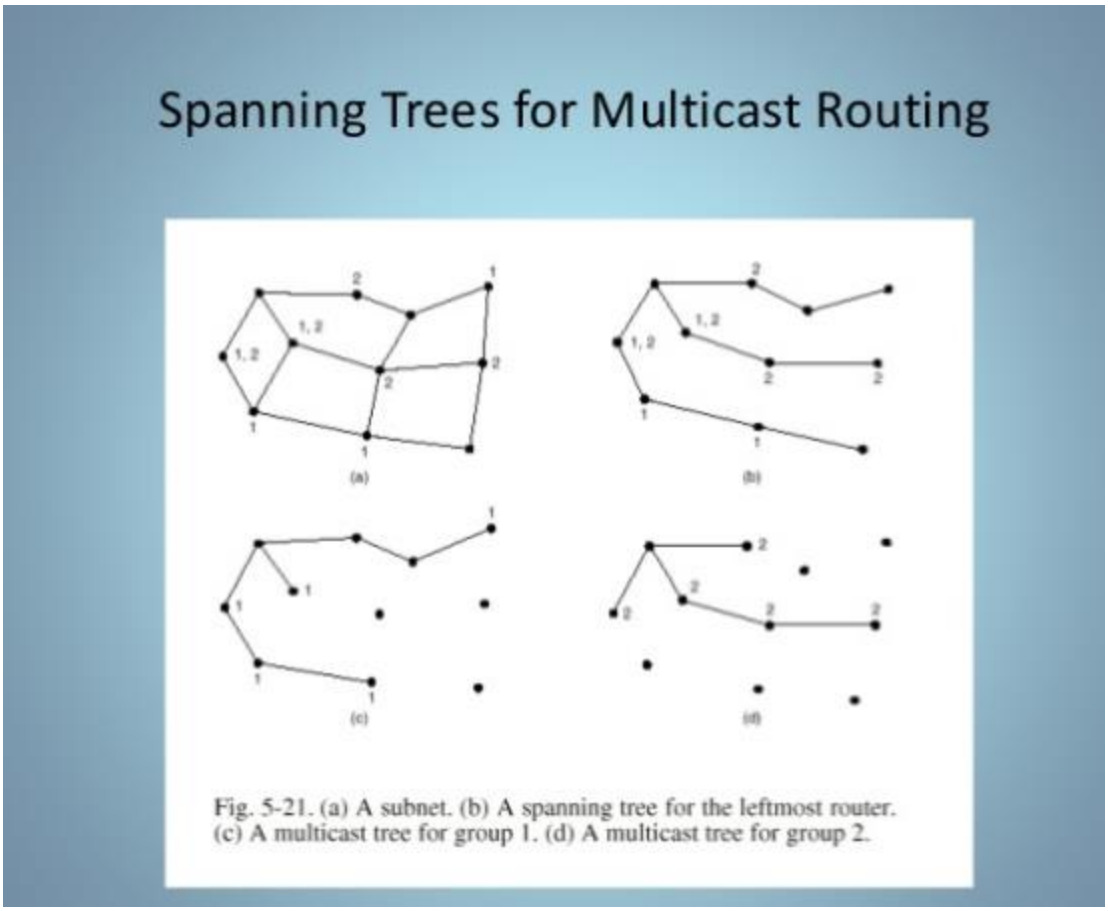
- Reverse Path Forwarding
  - Use When knowledge of a spanning tree is not available
  - Provides an approximation of spanning tree routing
  - Routers check to see if incoming packet arrives from the same line that the router uses to route outgoing packets to the broadcast source
    - If so, the router duplicates the packet on all other outgoing lines
    - Otherwise, the router discards the packet

tree built by reverse path forwarding



Multicast Routing

- A method to broadcast packets to well-defined groups
- Hosts can join multicast groups.
  - They inform their routers
  - Routers send group information throughout the subnet
- Each router computes a spanning tree for each group. The spanning tree includes all the routers needed to broadcast data to the group



Congestion Control

What is **congestion**?

A state occurring in network layer when the message traffic is so heavy that it slows down network response time.

Effects of Congestion

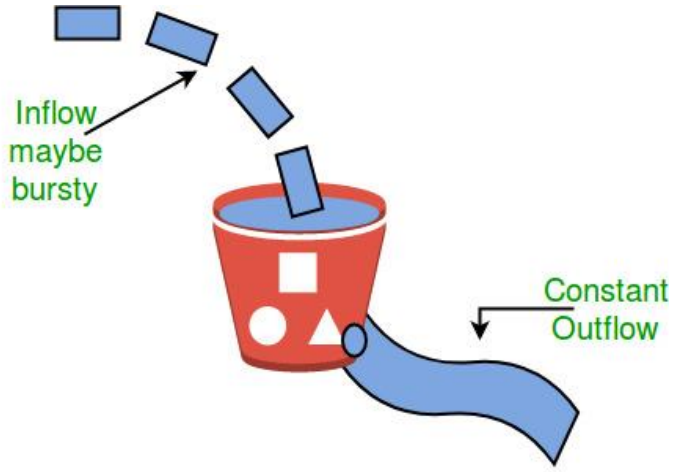
- As delay increases, performance decreases.
- If delay increases, retransmission occurs, making situation worse.

Congestion control algorithms

- **Leaky Bucket Algorithm**

Let us consider an example to understand

Imagine a bucket with a small hole in the bottom.No matter at what rate water enters the bucket, the outflow is at constant rate.When the bucket is full with water additional water entering spills over the sides and is lost.



Similarly, each network interface contains a leaky bucket and the following **steps** are involved in leaky bucket algorithm:

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

- **Token bucket Algorithm**

Need of token bucket Algorithm:-

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

Steps of this algorithm can be described as follows:

1. In regular intervals tokens are thrown into the bucket. *f*
2. The bucket has a maximum capacity. *f*
3. If there is a ready packet, a token is removed from the bucket, and the packet is sent.
4. If there is no token in the bucket, the packet cannot be sent.

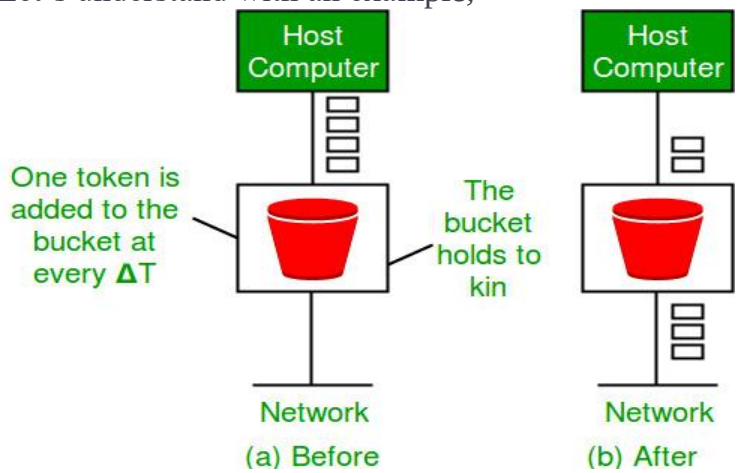
Let's understand with an example,

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

Ways in which token bucket is superior to leaky bucket:

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

Formula: M * s = C + ρ * s
where S – is time taken
M – Maximum output rate
ρ – Token arrival rate
C – Capacity of the token bucket in byte
Let's understand with an example,



| Quality of Service and Multimedia

Quality-of-Service (QoS) refers to traffic control mechanisms that seek to either differentiate performance based on application or network-operator requirements or provide predictable or guaranteed performance to applications, sessions or traffic aggregates. Basic phenomenon for QoS means in terms of packet delay and losses of various kinds.

Need for QoS –

- Video and audio conferencing require bounded delay and loss rate.
- Video and audio streaming requires bounded packet loss rate, it may not be so sensitive to delay.
- Time-critical applications (real-time control) in which bounded delay is considered to be an important factor.
- Valuable applications should be provided better services than less valuable applications.

QoS Specification –
QoS requirements can be specified as:

1. Delay
2. Delay Variation(Jitter)
3. Throughput
4. Error Rate

There are two types of QoS Solutions:

1. **Stateless Solutions –**
   Routers maintain no fine grained state about traffic, one positive factor of it is that it

is scalable and robust. But it has weak services as there is no guarantee about kind of delay or performance in a particular application which we have to encounter.

2. **Stateful Solutions –**
   Routers maintain per flow state as flow is very important in providing the Quality-of-Service i.e. providing powerful services such as guaranteed services and high resource utilization, provides protection and is much less scalable and robust.

Integrated Services(IntServ) –

1. An architecture for providing QoS guarantees in IP networks for individual application sessions.
2. Relies on resource reservation, and routers need to maintain state information of allocated resources and respond to new call setup requests.
3. Network decides whether to admit or deny a new call setup request.

IntServ QoS Components –

- Resource reservation: call setup signaling, traffic, QoS declaration, per-element admission control.
- QoS-sensitive scheduling e.g WFQ queue discipline.
- QoS-sensitive routing algorithm(QSPF)
- QoS-sensitive packet discard strategy.

RSVP-Internet Signaling –

It creates and maintains distributed reservation state, initiated by the receiver and scales for multicast, needs to be refreshed otherwise reservation times out as it is in soft state. Latest paths discovered through "PATH" messages (forward direction) and used by RESV messages (reserve direction).

Call Admission –

- Session must first declare it's QoS requirement and characterize the traffic it will send through the network.
- **R-specification:** defines the QoS being requested, i.e. what kind of bound we want on the delay, what kind of packet loss is acceptable, etc.
- **T-specification:** defines the traffic characteristics like bustiness in the traffic.
- A signaling protocol is needed to carry the R-spec and T-spec to the routers where reservation is required.
- Routers will admit calls based on their R-spec, T-spec and based on the current resource allocated at the routers to other calls.

Diff-Serv –

Differentiated Service is a stateful solution in which each flow doesn't means a different state. It provides reduced state services i.e. maintain state only for larger granular flows rather than end-to-end flows tries to achieve best of both worlds. Intended to address the following difficulties with IntServ and RSVP:

1. **Flexible Service Models:**
   IntServ has only two classes, want to provide more qualitative service classes: want to provide 'relative' service distinction.

2. **Simpler signaling:**
   Many applications and users may only want to specify a more qualitative notion of service.

Streaming Live Multimedia –

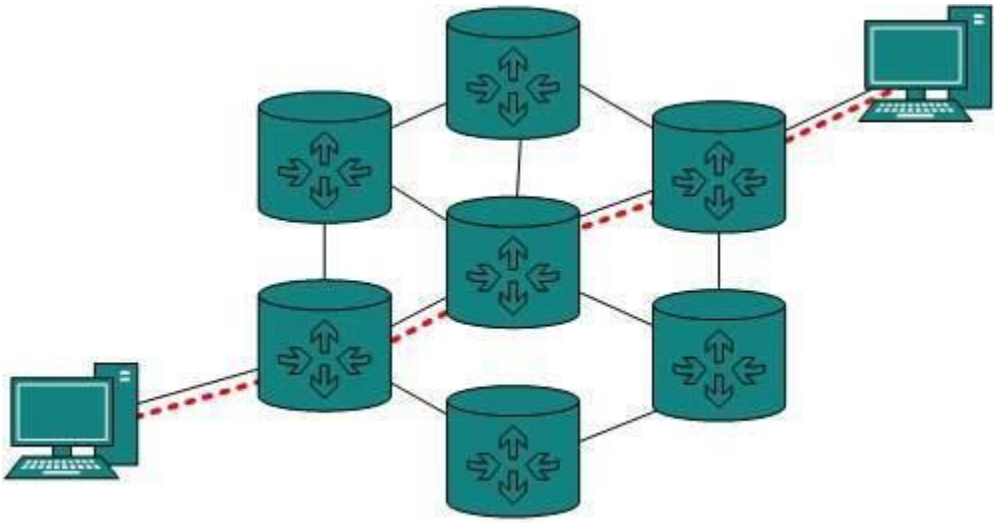
- **Examples:** Internet radio talk show, Live sporting event.
- **Streaming:** playback buffer, playback buffer can lag tens of seconds after and still have timing constraint.
- **Interactivity:** fast forward is impossible, but rewind and pause is possible.

Internetworking in Computer Network

In real world scenario, networks under same administration are generally scattered geographically. There may exist requirement of connecting two different networks of same kind as well as of different kinds. Routing between two networks is called internetworking.

Networks can be considered different based on various parameters such as, Protocol, topology, Layer-2 network and addressing scheme.

In internetworking, routers have knowledge of each other's address and addresses beyond them. They can be statically configured go on different network or they can learn by using internetworking routing protocol.

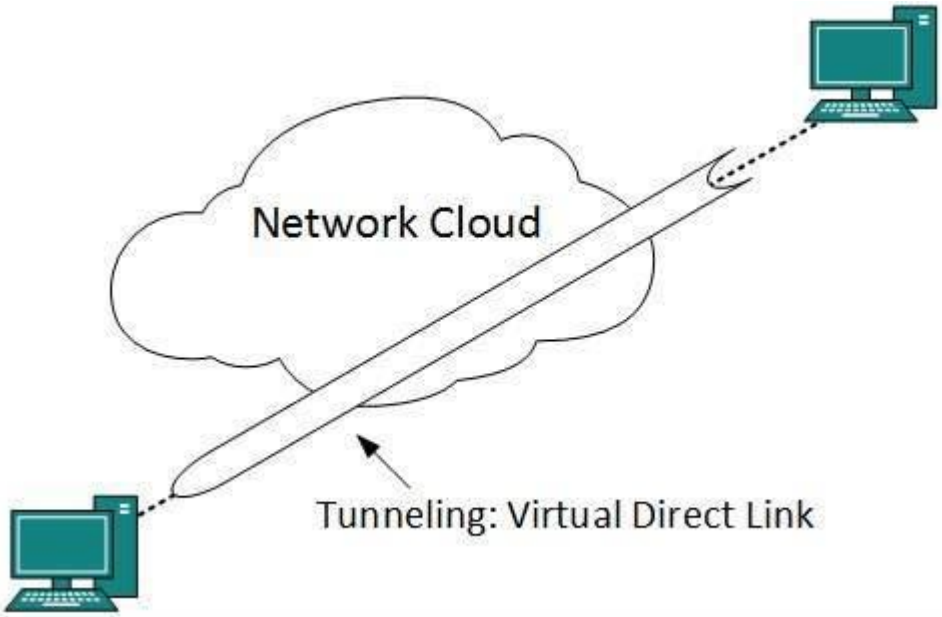


Routing protocols which are used within an organization or administration are called Interior Gateway Protocols or IGP. RIP, OSPF are examples of IGP. Routing between different organizations or administrations may have Exterior Gateway Protocol, and there is only one EGP i.e. Border Gateway Protocol.

Tunneling

If they are two geographically separate networks, which want to communicate with each other, they may deploy a dedicated line between or they have to pass their data through intermediate networks.

Tunneling is a mechanism by which two or more same networks communicate with each other, by passing intermediate networking complexities. Tunneling is configured at both ends.



When the data enters from one end of Tunnel, it is tagged. This tagged data is then routed inside the intermediate or transit network to reach the other end of Tunnel. When data exists the Tunnel its tag is removed and delivered to the other part of the network.

Both ends seem as if they are directly connected and tagging makes data travel through transit network without any modifications.

Packet Fragmentation

Most Ethernet segments have their maximum transmission unit (MTU) fixed to 1500 bytes. A data packet can have more or less packet length depending upon the application. Devices in the transit path also have their hardware and software capabilities which tell what amount of data that device can handle and what size of packet it can process.

If the data packet size is less than or equal to the size of packet the transit network can handle, it is processed neutrally. If the packet is larger, it is broken into smaller pieces and then forwarded. This is called packet fragmentation. Each fragment contains the same destination and source address and routed through transit path easily. At the receiving end it is assembled again.

If a packet with DF (don't fragment) bit set to 1 comes to a router which can not handle the packet because of its length, the packet is dropped.

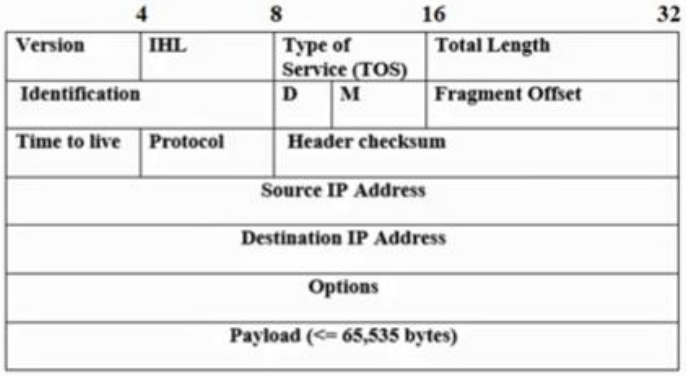
When a packet is received by a router has its MF (more fragments) bit set to 1, the router then knows that it is a fragmented packet and parts of the original packet is on the way.

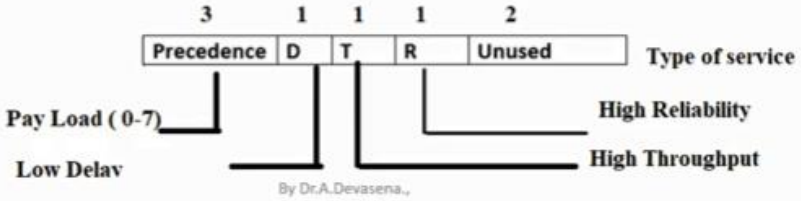
If packet is fragmented too small, the overhead is increases. If the packet is fragmented too large, intermediate router may not be able to process it and it might get dropped.

Network layer in the Internet

- **The Internet Protocol provides the basis for the interconnections of internet.**
- **IP is a datagram protocol and its packets contain an IP header.**

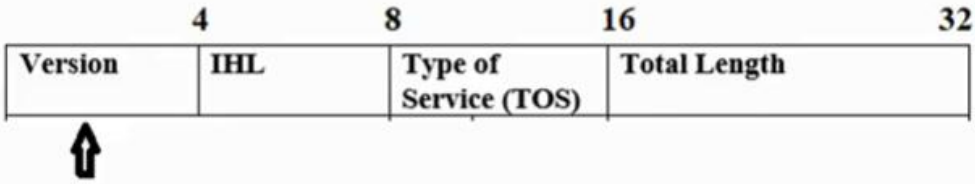
The basic header without options is shown as following.





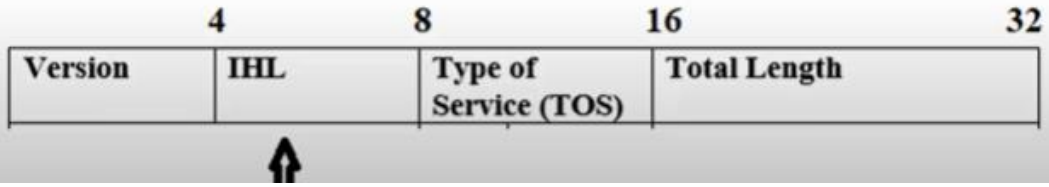
Version field

- The version field contains the version of IP,
- IPv4 or IPv6



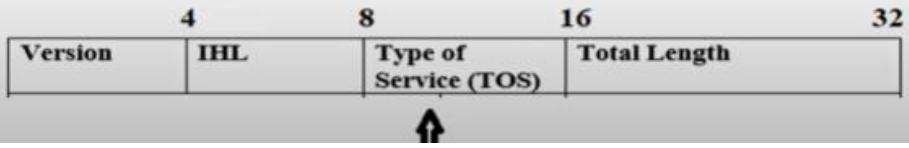
Internet Header Length (IHL) field

- **The IHL field specifies the actual length of the header in multiples of 32 bit words.**
- **The minimum length is 5 and maximum length is 15.**



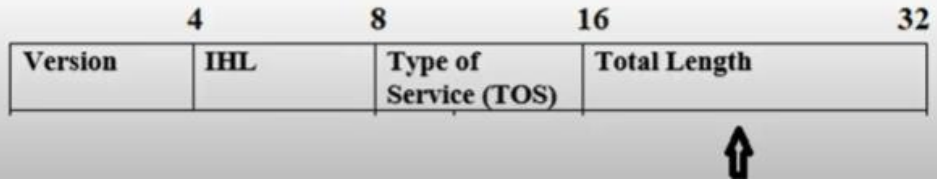
Type of Service (TOS) field

- **The type of service field allows an application protocol/process to specify the relative priority of the application data and the preferred attributes associated with the path to be followed.**
- **It is used by each gateway and router during the transmission and routing packet to transmit packets of higher priority first.**
- **It is also used to select a line/route that has the specified attributes should be available.**



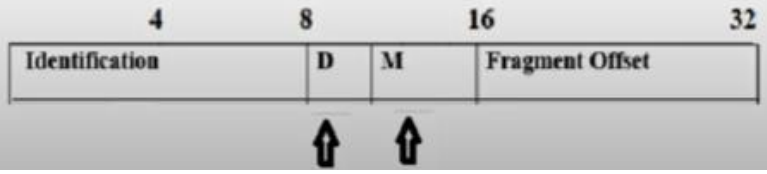
Total Length field

- The total length field defines the total length of the initial datagram including the header and payload parts.
- When the contents of the initial datagram used to be transferred is multiple packets then the value in this field is used by the destination host to reassemble the payload contained within each packet is shown as fragment.



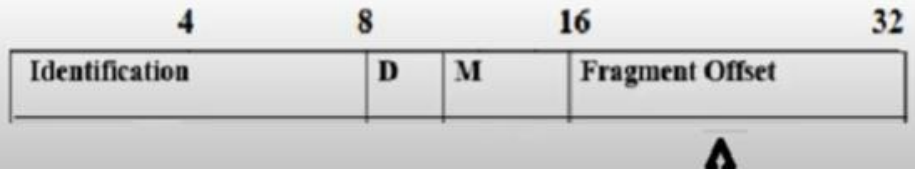
D-bit , M-Bit field

- D- bit: Don't fragment or D-bit is set by a source host and is examined by routers. It indicates that the packet should not be fragmented.
- M-bit: More fragment or M-bit is used during the reassembly procedure associated with data transfer involving multiple smaller packets/fragments. It is set to 1 for all but for last packet/fragment will set as 0.



Fragment offset field

- It is used to indicate the position of the first byte of the fragment contained within a smaller packet involving multiple smaller packets/fragments.
- It is set to 1 for all but for last packet/fragment will set as 0.

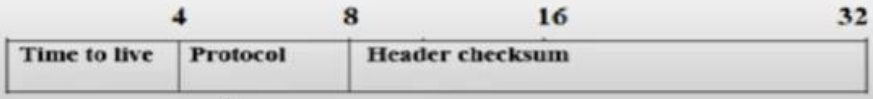


Time to live field

- Time to live field defines the maximum for which a packet can be in transit across the internet.
- The value of time to live is seconds and it is set by the IP in the source host.
- It is determined by each gateway and router by a defined amount and it becomes zero if the packet is discarded.

Protocol field

- **IP Protocol field is used to enable the destination IP to pass the payload within each received packet to the same(peer) protocol that sent the data.**
- **This can be internal network layer protocol such as the ICMP or a higher layer protocol such as TCP or UDP.**



Header checksum field

- The header checksum can be applied to the header part of the datagram and it is safe-guard against corrupted packets being routed to incorrect destinations.

Source and Destination IP address

- **The source and destination IP address indicate the sending host and the intended recipient host for this diagram.**