UNIT – 3

Network Layer

What is Network Layer?

- The basic function of **Network layer is to provide an end to end communication** capability to the transport layer which lies above it at sender level similarly receiver level.
- The **network layer holds the responsibility of managing subnet performance.**
- **This layer is more focused to control the operations of data transmission, routing and switching technologies, packet forwarding and sequencing.**
- To achieve the goal the network layer must know the **Topology of the communication Subnet** i.e **set of all routers and choose appropriate path through it.**
- The **Network layer protocols are concern exchange of packets of information** between transport layer entities.
- A **packet is a group of bits (i.e collection of frames).**
- The **services provided by the network layer to transport layer is called Network services.**
- It **makes routing of data through network from source to destination.**

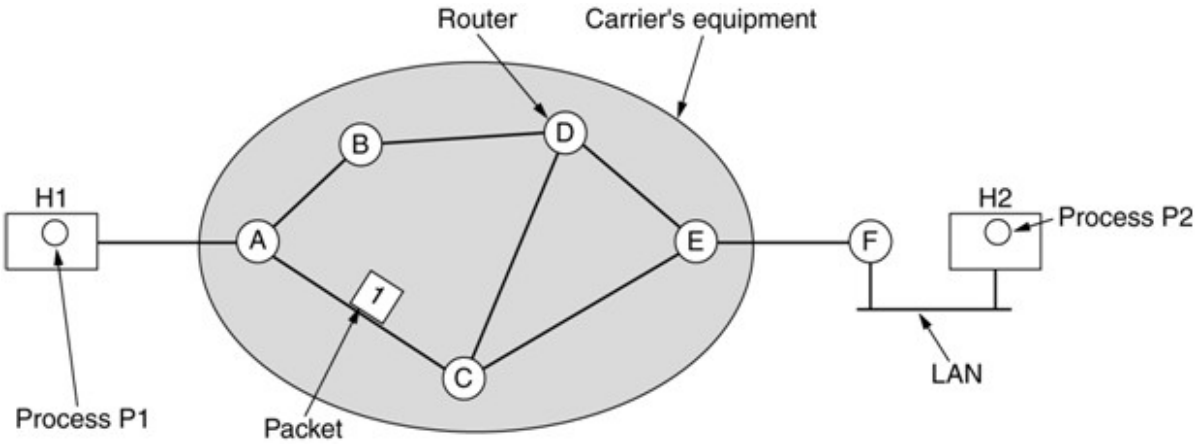
Design Issues of Network Layer:

- Store and Forward Packet switching
- Services Provided to the Transport Layer
- Implementation of Connection oriented Service
- Implementation of Connectionless Service
- Comparison of Datagram (connection less)and Virtual-circuit (connection oriented) subnets

Store and Forward Packet switching:

- **Packet switching is a method of transferring the data to a network in form of packets.**
- **In order to transfer the file fast and efficient manner over the network** and **minimize the transmission latency**(delay), **the data is broken into small pieces of variable length, called Packet**.
- **At the destination, all these small-parts (packets) has to be reassembled,** belonging to the same file.
- Packet Switching uses **Store and Forward technique** while switching the **packets;** while **forwarding the packet each hop first store that packet then forward.**
- This technique is very beneficial because packets may get discarded at any hop due to some reason.
- **Each router verifies possible number of links which link is better.**

- **More than one path is possible between a pair of source and destination.**
- **Each packet contains Source and destination address** using which they **independently travel through the network.**
- In other words, **packets belonging to the same file may or may not travel through the same path.**



- Host H1 is directly connected to one of the carrier's routers, A, by a leased line.
- In contrast, H2 is on a LAN with a router, F, owned and operated by the customer.
- This router also has a leased line to the carrier's equipment. We have shown F as being outside the oval because it does not belong to the carrier, but in terms of construction, software, and protocols, it is probably no different from the carrier's routers.
- Whether it belongs to the subnet is arguable, routers on customer premises are considered part of the subnet because they run the same algorithms as the carrier's routers (and our main concern here is algorithms).

➤ Services Provided to the Transport Layer:

- The **network layer provides services to the transport layer at the network layer/transport layer interface.**
- An important question is what kind of services the network layer provides to the transport layer.
- The network layer services have been designed with the following goals in mind.

1. **The services should be independent of the router technology.**

   2. The transport layer should be shielded from the number, type, and topology of the routers present.

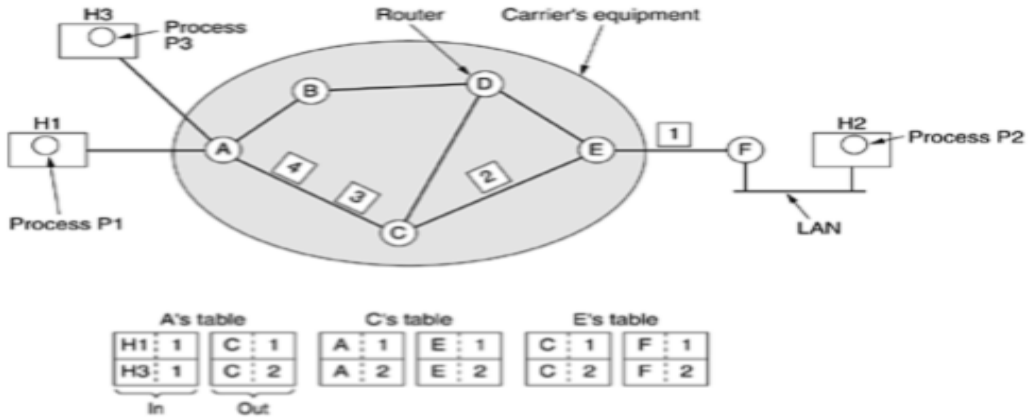
   3. The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

- Given these goals, the designers of the network layer have a lot of freedom in writing detailed specifications of the services to be offered to the transport layer.
- whether the network layer should provide **connection-oriented service or connectionless service.**

➤ Implementation of Connection oriented Service:

- connection-oriented service, we need a **virtual-circuit** subnet.
- The **virtual circuits is to avoid having to choose a new route for every packet sent** .
- All the packets are transmitted depends on **same path.**
- Instead, **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, exactly the same way that the telephone system works.**
- **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.

Figure 5-3. Routing within a virtual-circuit subnet.



Example:

consider the situation of Fig. Here, host H1 has established connection 1 with host H2. It is remembered as the first entry in each of the routing tables. 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.

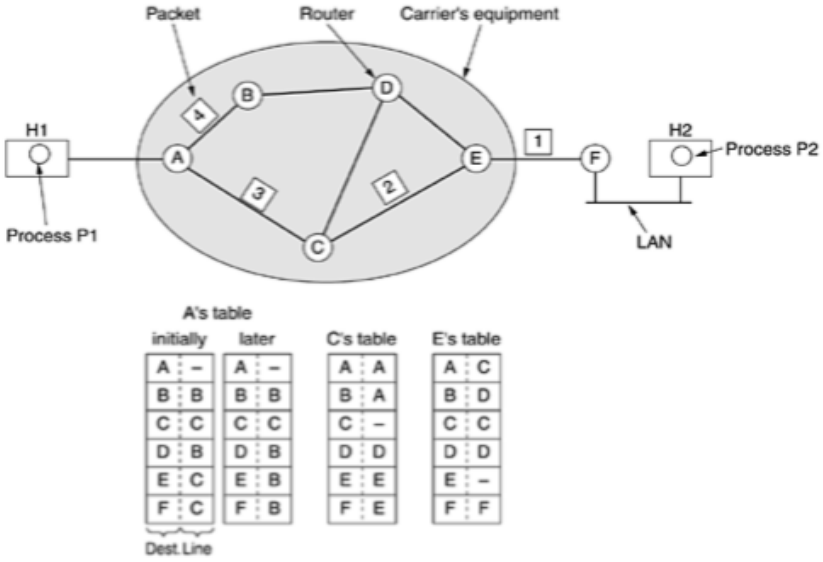
Now let us consider what happens if H3 also wants to establish a connection to H2. It chooses connection identifier 1 (because it is initiating the connection and this is its only connection) and tells the subnet to establish the virtual circuit. This leads to the second row in the tables. Note that 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. In some contexts, this is called label switching.

➤ Implementation of Connectionless Service:

- Connection less services is also called as **Datagrams.**

- **Datagram is a <u>self contained message unit which contains sufficient information to allow it to be routed from source to destination.</u>**
- *Each packet is treated as independently.*
- *The network layer simply accepts messages from the transport layer and sends out to the destination.*
- *Datagram system is to postal system.*
- **Each packet contains the full source and destination addresses .**
- **Each packet follows different routes from source to destination and they do not necessarily arrive in the same order. As** *they transmitted i.e No sequence and flow control is done at receiver.*

Figure 5-2. Routing within a datagram subnet.



➤ ***Comparison of Datagram (connection less)and Virtual-circuit (connection oriented) subnets:***

Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Comparison of datagram and virtual-circuit networks

Routing Algorithms :

- **Routing** is process of establishing the routes that data packets must follow to reach the destination.
- In this process, a **routing table is created which contains information regarding routes which data packets follow.**
- **Various routing algorithm are used for the purpose of deciding which route an incoming data packet needs to be transmitted on to reach destination efficiently.**

Classification of Routing Algorithms: The routing algorithms can be classified as follows:

1. Adaptive Algorithms .
2. Non Adaptive Algorithms .

1. Adaptive Algorithms :

- These are the algorithms which **change their routing decisions whenever network topology or traffic load changes.**
- The **changes in routing decisions are reflected in the topology as well as traffic of the network.**
- Also known as **dynamic routing**, these make use of dynamic information such as current topology, load, delay, etc. to select routes.
- **Optimization parameters are distance, number of hops and estimated transit time.**

2. Non-Adaptive Algorithms :

- These are the algorithms which **do not change t**heir routing decisions once they have been selected.
- This is also known as **static routing** as route to be taken is computed in advance and downloaded to routers when router is booted.

➤ Shortest Path Algorithm (Least cost Algorithm) : (Dijkstras Algorithm)

Several algorithms for **computing the shortest path between two nodes of a graph** are known. This one is due to **Dijkstras** (1959).

- **Each node is labeled (in parentheses) with its distance from the source node along the best known path.**
- **Initially, no paths are known, so all nodes are labeled with infinity.**
- **As the algorithm proceeds and paths are found, the labels may change, reflecting better paths.**
- **A label may be either tentative or permanent.**
- **Initially, all labels are tentative.** When it is discovered that a label represents the shortest possible path from the source to that node, it is made permanent and never changed thereafter.

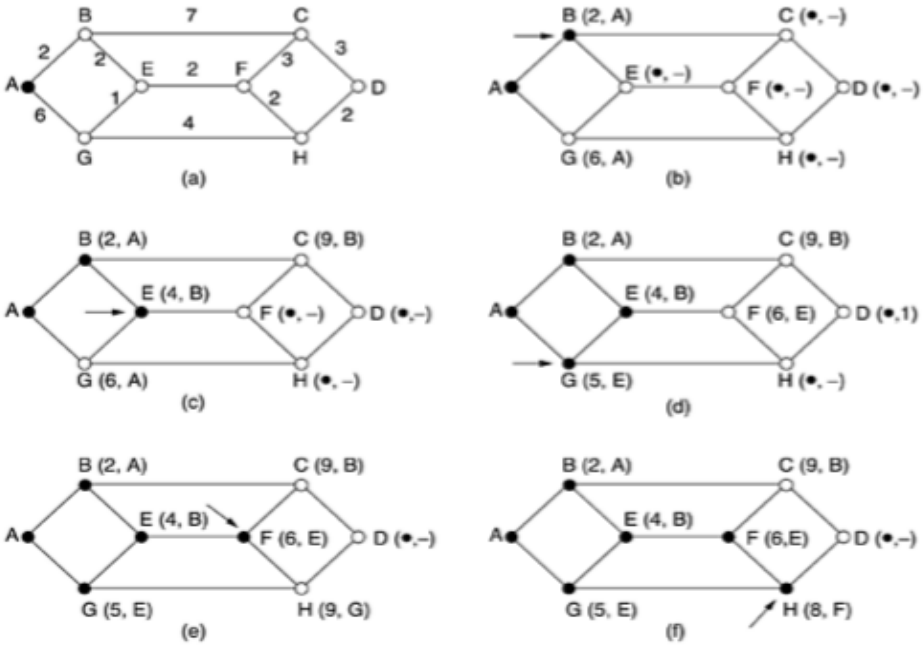
Routing algorithms with a simple technique for computing optimal paths which gives a complete picture of the network and are able to form distributed routing algorithm to find, even though not all routers may know all of the details of the network.

The idea 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.** To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.

- **In shortest path routing algorithm the path length between each node is measured either in Distance , Bandwidth , Average Traffic , Communication Cost or Delay** etc...
- This **Algorithm computes shortest path** according to any one of a number of criteria or a combination of criteria.
- **For this a Graph of subnet is drawn** . With **each node of a graph is representing a router , and each Arc of a graph is representing a communication link. Each link Associated with cost.**
- **To choose route b/w a given pair of routers , this algorithm's just find shortest path b/w them on the graph.**

Figure 5-7.

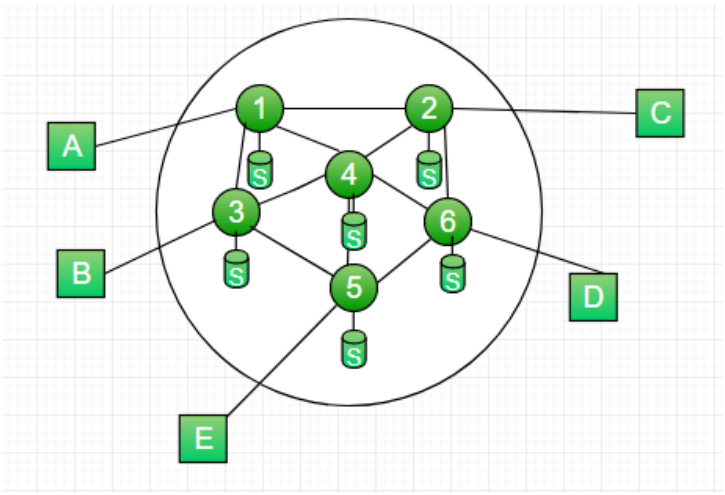
The first five steps used in computing the shortest path from A to D. The arrows indicate the working node.



➤ FLOODING:

A simple local technique in which every incoming packet is sent out on every outgoing line except the one it arrived on is called flooding.

- **Flooding generates large number of duplicate packets.** One way to present this is for each node to renumber the identity of the packets it has already sent.
- One such measure is to have a hop counter contained in the header of each packet, **which is decremented at each hop. When the count reaches zero, the packet is discarded.** The counter is set to maximum value i.e diameter of subnet.
- **No** routing table maintained.
- Always gives shortest path.
- It is more reliable.
- Traffic is high.
- Duplicate packets are present.



Characteristics –

- *All possible routes between Source and Destination is tried.*
- *A packet will always get through if path exists*
- *As all routes are tried, there will be at least one route which is the shortest*
- *All nodes directly or indirectly connected are visited.*

Limitations –

- *Flooding generates vast number of duplicate packets.*
- *Suitable damping mechanism must be used.*

Hop-Count –

- *A hop counter may be contained in the packet header which is decremented at each hop, with the packet being discarded when the counter becomes zero.*
- *The sender initializes the hop counter. If no estimate is known, it is set to the full diameter of the subnet.*
- *Keep track of the packets which are responsible for flooding using a sequence number. Avoid sending them out a second time.*

➤ Hierarchical Routing:

* **When the network size grows the number of routers in the network increases.**

* *Consequently the* **size of routing tables as well and routers cannot handle network traffic as efficiently.**

* *We use hierarchical routing to overcome this problem.*

* Hierarchical Routing is essentially **Divide and Conquer strategy,**

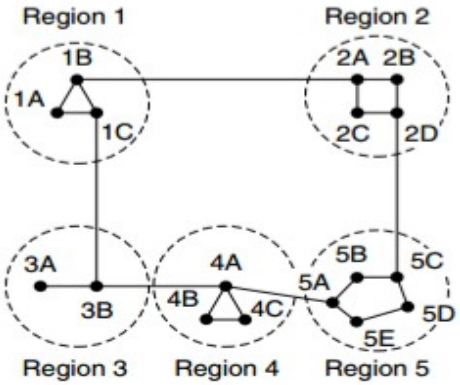
* The **network is divided into different regions and a router for a particular region knows only about its own domain and other neighbor routers.**

* In **hierarchical routing routers are classified in to groups known as regions.**

* **Each router has only the information about the routers in its own region and has no information about routers in the other regions .**

* **So routers just save one record in their table for every other region.**

- **Router just save one record in their table for every other region.**
- **Router only contains the record of their immediate neighbors in the table.**
- **In three level hierarchical routing the network is called into a number of clusters.**
- **Each cluster is made of a number of regions and each region contains a number or routers.**
- **In this method it will route first to the region then to the IP Prefix within the region- hide details within a region from outside of the region.**



(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)

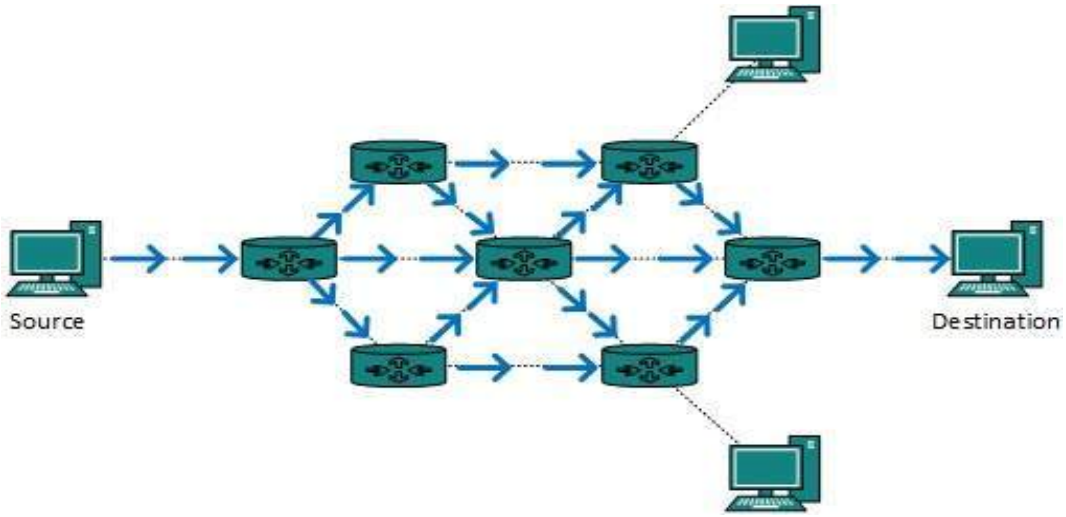
Figure 5-14. Hierarchical routing.

Broadcast Routing:

Sending a packet to all destinations simultaneously is called broadcasting.

Ex: a service distributing weather reports, stock market updates, or live radio programs might work best by sending to all machines and letting those that are interested read the data.

- *Various methods are proposed but some are not in practice. for* ***example*** *sending the data to each destination in a network which is widely applicable but wasteful of bandwidth and source must have the complete list of all destinations.*
- ***Multi destination routing****: In multi-destination routing, each packet contains either 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.* ***This is also not made in practice because of bandwidth and destination list problems.***



- *The idea for* ***reverse path forwarding*** *is elegant and remarkably simple once it has been pointed out. 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, 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 otherwise it is accepted.*

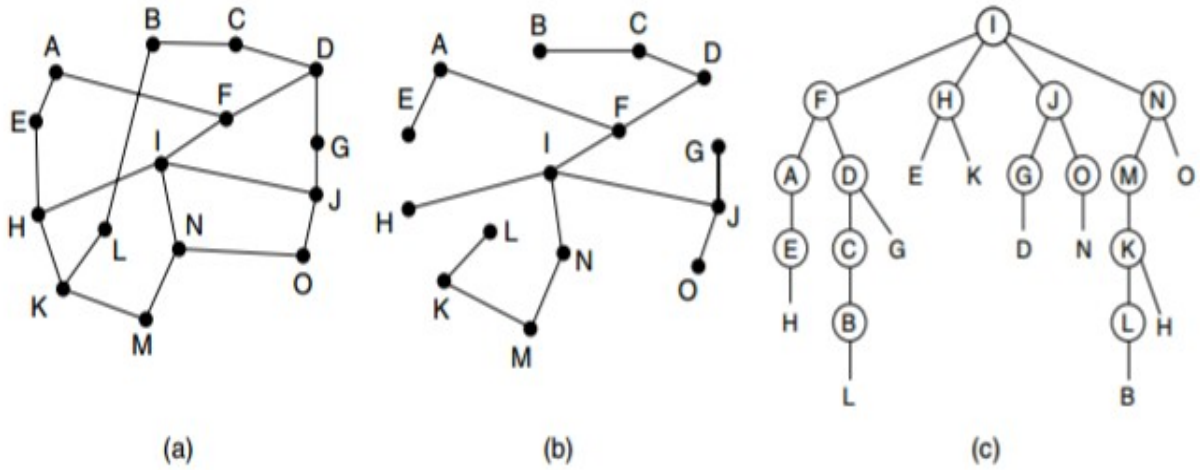


Figure 5-15. Reverse path forwarding. (a) A network. (b) A sink tree. (c) The tree built by reverse path forwarding.

➤ Multicast Routing:

- **Sending a message to well-defined groups** that are numerically large in size but small compared to the network as a whole is called multicasting, and the routing algorithm used is called multicast routing.
- **All multicasting schemes require some way to create and destroy groups and to identify which routers are members of a group.** How these tasks are accomplished is not of concern to the routing algorithm.

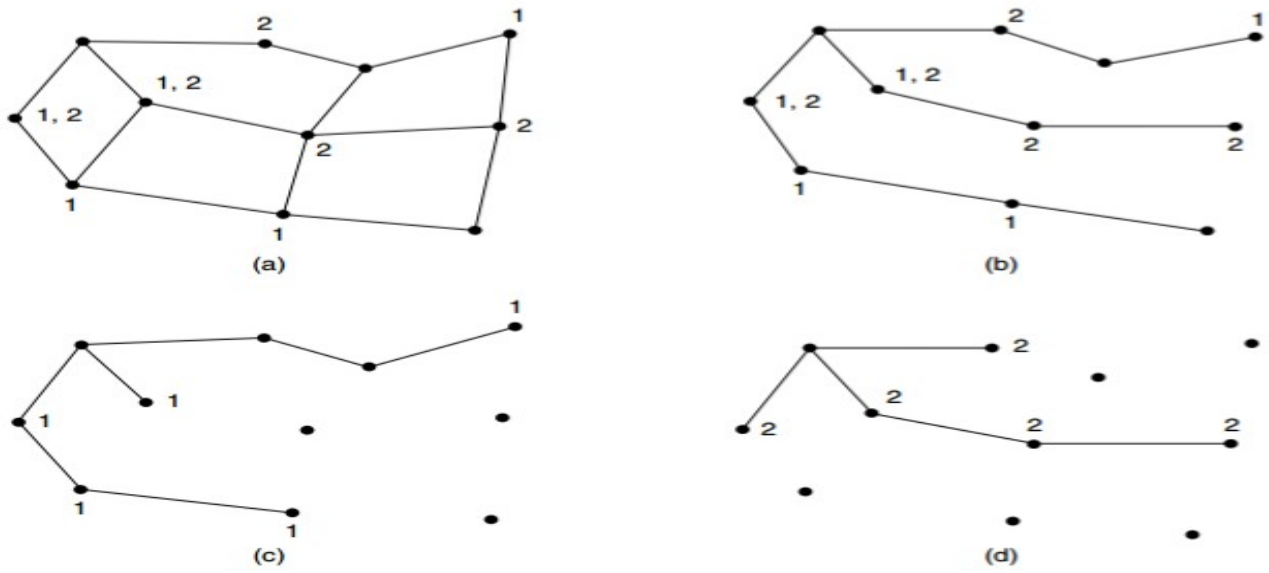


Figure 5-16. (a) A network. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1. (d) A multicast tree for group 2.

Example:

Consider the two groups, 1 and 2, in the network shown in Fig. 5-16(a). Some routers are attached to hosts that belong to one or both of these groups, as indicated in the figure.

A spanning tree for the leftmost router is shown in Fig. 5-16(b). This tree can be used for broadcast but is overkill for multicast, as can be seen from the two pruned versions that are shown next.

In Fig. 5-16(c), all the links that do not lead to hosts that are members of group 1 have been removed. The result is the multicast spanning tree for the leftmost router to send to group 1. Packets are forwarded only along this spanning tree, which is more efficient than the broadcast tree because there are 7 links instead of 10. Fig. 5-16(d) shows the multicast spanning tree after pruning for group 2. It is efficient too, with only five links this time. It also shows that different multicast groups have different spanning trees.

➤ Distance-vector routing protocol :(Bell man ford algorithm)

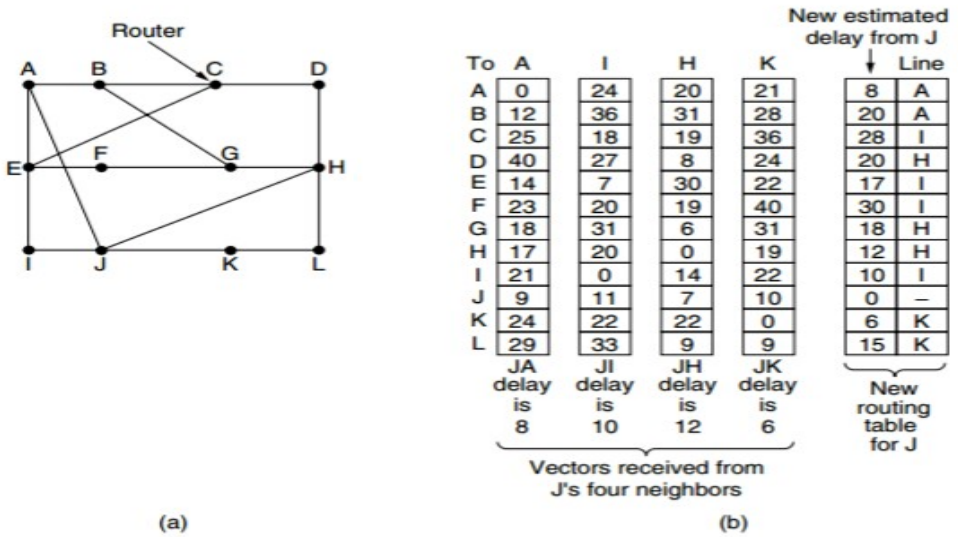
Distance-vector routing protocol in data networks determines the best route for data packets based on distance.

- Distance-vector routing protocols measure the distance by the number of routers a packet has to pass, one router counts as one hop.
- Some distance-vector protocols also take into account network latency (Best amount of time to transmission of data) and other factors that influence traffic on a given route.
- To determine the best route across a network, routers, on which a distance-vector protocol is implemented, exchange information with one another, usually routing tables plus hop counts for destination networks and possibly other traffic information.
- Distance-vector routing protocols also require that a router informs its neighbors of network topology changes periodically.

Distance-vector routing protocols use the Bellman–Ford algorithm and Ford–Fulkerson algorithm to calculate the best route.

Another way of calculating the best route across a network is based on link cost, and is implemented through link-state routing protocols.

➤ 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 for that destination and an estimate of the distance to that destination.



To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	—
K	24	22	22	0	6	K
L	29	33	9	9	15	K

Figure 5-9. (a) A network. (b) Input from *A*, *I*, *H*, *K*, and the new routing table for *J*.

Example: JA = 8

JI = 10

JH =12

JK = 6

- J ----> A

JA=8
JI + IA = 10 +24 =34
JH + HA = 12 +20 =32
JK + KA =6 + 21 =27

- J -----> B

JA + AB =8 + 12 =20
JI + IB = 10 +36 =46
JH + HB = 12 + 31 =43
JK + KB = 6+ 28 = 34

Example: JA = 8

JI = 10

JH =12

JK = 6

- J ----> A

JA=8
JI + IA = 10 +24 =34
JH + HA = 12 +20 =32
JK + KA =6 + 21 =27

- J -----> B

JA + AB =8 + 12 =20
JI + IB = 10 +36 =46
JH + HB = 12 + 31 =43
JK + KB = 6+ 28 = 34

➤ The Count-to-Infinity Problem:

The settling of routes to best paths across the network is called ***convergence.*** *Distance vector routing is useful as a simple technique by which routers can collectively compute shortest paths, but it has a serious drawback in practice: although it converges to the correct answer, it may do so slowly.*

A	B	C	D	E	
•	•	•	•	•	Initially
1	•	•	•	•	After 1 exchange
1	2	•	•	•	After 2 exchanges
1	2	3	•	•	After 3 exchanges
1	2	3	4	•	After 4 exchanges

(a)

A	B	C	D	E	
•	1	2	3	4	Initially
3	1	2	3	4	After 1 exchange
3	3	4	3	4	After 2 exchanges
5	5	4	5	4	After 3 exchanges
5	5	6	5	6	After 4 exchanges
7	7	6	7	6	After 5 exchanges
7	7	8	7	8	After 6 exchanges
•	•	•	•	•	

(b)

Figure 5-10. The count-to-infinity problem.

➤ *To initiate a vector exchange at all routers simultaneously. At the time of the first exchange, B learns that its left-hand neighbor has zero delay to A. B now makes an entry in its routing table indicating that A is one hop away to the left. All the other routers still think that A is down. At this point, the routing table entries for A are as shown in the second row of Fig.(a).*

➤ *On the next exchange, C learns that B has a path of length 1 to A, so it updates its routing table to indicate a path of length 2, but D and E do not hear the good news until later. Clearly, the good news is spreading at the rate of one hop per*

exchange. In a network whose longest path is of length N hops, within N exchanges everyone will know about newly revived links and routers.

- In fig.b, Routers B, C, D, and E have distances to A of 1, 2, 3, and 4 hops, respectively. Suddenly, either A goes down or the link between A and B is cut.
- At the first packet exchange, B does not hear anything from A. B suspect that C's path runs through B itself. For all B knows, C might have ten links all with separate paths to A of length 2. As a result, B thinks it can reach A via C, with a path length of 3. D and E do not update their entries for A on the first exchange.
- On the second exchange, C notices that each of its neighbors claims to have a path to A of length 3. It picks one of them at random and makes its new distance to A 4, as shown in the third row of Fig.(b).
- no router ever has a value more than one higher than the minimum of all its neighbors. Gradually, all routers work their way up to infinity, but the number of exchanges required depends on the numerical value used for infinity. this problem is known as the **count-to-infinity** problem.

Congestion Control in Computer Networks:

What is **congestion**?

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

(or)

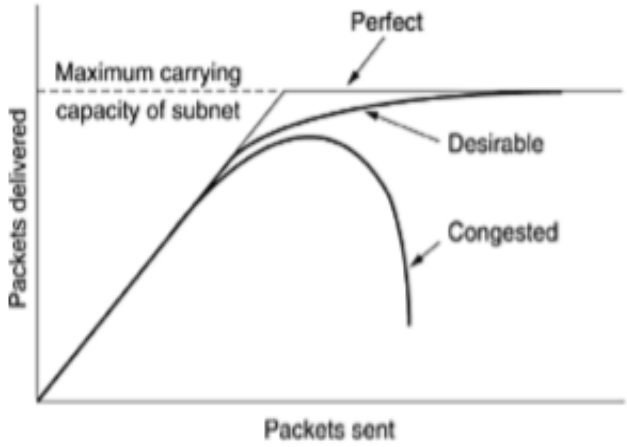
When too many packets are present in (a part of) the subnet, performance degrades. This situation is called congestion

- **Congestion control** refers to the techniques used to control or prevent congestion.
- **Congestion control** is the process of maintaining the number of packets in the network below a certain level at which performance falls off.

Effects of Congestion

- As delay increases, performance decreases.
- If delay increases, retransmission occurs, making situation worse.
- When the number of packets dumped into the subnet by the hosts is within its carrying capacity, they are all delivered (except for a few that are afflicted with transmission errors) and the **number delivered is proportional to the number sent.**
- However, as **traffic increases too far**, the routers are no longer able to cope and **they begin losing packets.** This tends to make matters **worse.** At **very high trafffic, performance collapses completely and almost no packets are delivered.**

When too much traffic is offered, congestion sets in and performance degrades sharply.



Causes Of Congestion : *(Congestion can be brought on by several factors)*

- *If all of a sudden,* ***streams of packets begin arriving on three or four input lines and all need the same output line****, a queue will build up.*
- *If there is insufficient memory to hold all of them, packets will be lost.*
- ***Slow processors can also cause congestion****. If the routers' CPUs are slow at performing the bookkeeping tasks required of them (queueing buffers, updating tables, etc.), queues can build up, even though there is excess line capacity.*
- *Similarly, low-bandwidth lines can also cause congestion. Upgrading the lines but not changing the processors, or vice versa.*
- *The band width of the links are also important in congestion . The links to be used must be high bandwidth to avoid the congestion.*
- *When the arrival of the packets are not uniform i.e when the traffic is bursty.*

General Principles of Congestion Control :

This approach leads to dividing all solutions into two groups:

- ***Open loop Congestion Control.***
- ***Closed loop Congestion Control.***

Open Loop Congestion Control :

Open loop congestion control policies are applied to prevent congestion before it happens. The

congestion control is handled either by the source or the destination.

Closed Loop Congestion Control:

Closed loop congestion control technique is used to treat or alleviate congestion after it happens. Several techniques are used by different protocols.

This approach has three parts when applied to congestion control:

1. **Monitor the system to detect when and where congestion occurs.**
2. **Pass this information to places where action can be taken.**
3. **Adjust system operation to correct the problem.**

Congestion Prevention Policies :

➤ Approaches to Congestion Control:

Here we have two solutions to handle the presence of congestion where the load is greater than resource:

1. Increase the resource
2. Decrease the load

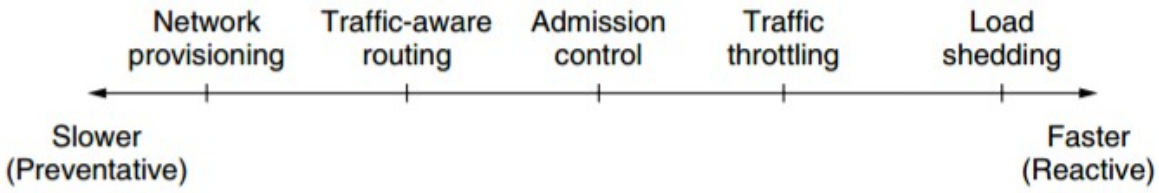


Figure 5-22. Timescales of approaches to congestion control.

1. The most basic way to avoid congestion is to build a network that is well matched to the traffic it carries. Sometimes resources can be added dynamically when there is serious congestion, for example, turning on spare routers or enabling lines that are normally used only as backups or purchasing bandwidth on the open market. More often, links and routers that are regularly heavily utilized are upgraded at the earliest opportunity. This is called **provisioning** and happens on a time scale of months, driven by long-term traffic trends.

2. Routes can be tailored to traffic patterns that change during the day as network users wake and sleep in different time zones, so they used to shift traffic away from heavily used paths by changing the shortest path weights. This is called **traffic-aware routing**. Splitting traffic across multiple paths is also helpful.

3. However, sometimes it is not possible to increase capacity. The only way then to beat back the congestion is to decrease the load. In a virtual-circuit network, new connections can be refused if they would cause the network to become congested. This is called **admission control.**

4. Finally, when all else fails, the network is forced to discard packets that it cannot deliver. The general name for this is **load shedding.** A good policy for choosing which packets to discard can help to prevent congestion collapse.

Traffic-Aware Routing:

These schemes adapted to changes in topology, but not to changes in load .The goal in taking load into account when computing routes is to shift traffic away from hotspots that will be the first places in the network to experience congestion.

1. The most direct way to do this 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. Least-weight paths will then favor paths that are more lightly loaded, all else being equal.

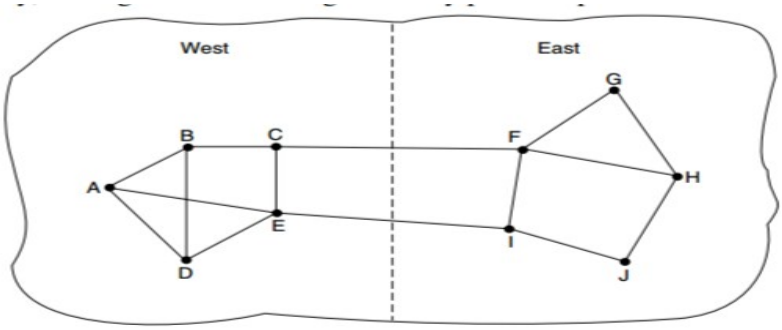


Figure 5-23. A network in which the East and West parts are connected by two links.

2. Consider the network of Fig. 5-23, which is divided into two parts, East and West, connected by two links, CF and EI.

3. 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.

4. After the new routing tables have been installed, most of the East-West traffic will now go over EI, loading this link. Consequently, CF will appear to be the shortest path. As a result, the routing tables may oscillate wildly, leading toerratic routing and many potential problems.

Admission Control:

One technique that is widely used in virtual-circuit networks to keep congestion at bay is admission control. The idea is simple: do not set up a new virtual circuit unless the network can carry the added traffic without becoming congested.

Admission control can also be combined with traffic-aware routing by considering routes around traffic hotspots as part of the setup procedure.

For example, consider the network illustrated in Fig. 5-24(a), in which two routers are congested, as indicated 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. 5-24(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.

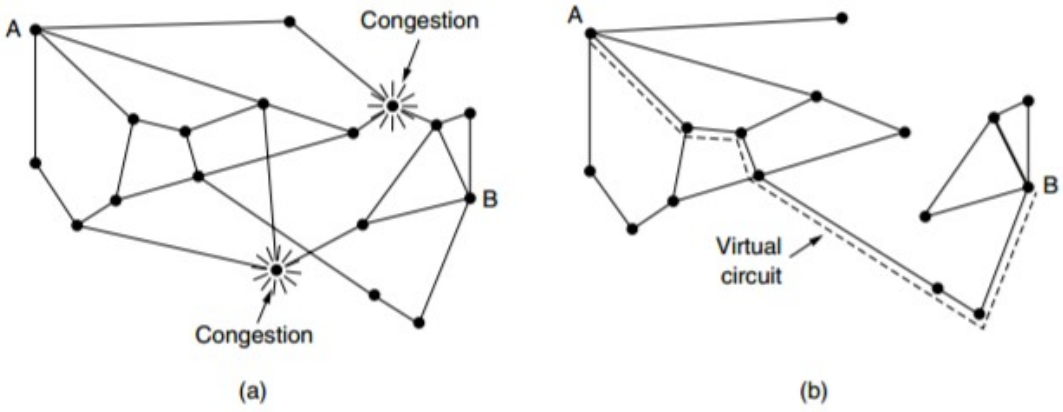


Figure 5-24. (a) A congested network. (b) The portion of the network that is not congested. A virtual circuit from *A* to *B* is also shown.

➤ Traffic Throttling:

1. Let us now look at some approaches to throttling traffic that can be used in both datagram networks and virtual-circuit networks. Each approach must solve two problems. First, routers must determine when congestion is approaching, ideally before it has arrived. To do so, each router can continuously monitor the resources it is using.
2. Three possibilities are the utilization of the output links, the buffering of queued packets inside the router, and the number of packets that are lost due to insufficient buffering. Of these possibilities, the second one is the most useful.

➤ Choke Packets:

1. A more direct way of telling the source to slow down.
2. A choke packet is a control packet generated at a congested node and transmitted to restrict traffic flow.
3. The source, on receiving the choke packet must reduce its transmission rate by a certain percentage. 4. An example of a choke packet is the ICMP Source Quench Packet.

➤ Hop-by-Hop Choke Packets

1. Over long distances or at high speeds choke packets are not very effective.
2. A more efficient method is to send to choke packets hop-by-hop.
3. This requires each hop to reduce its transmission even before the choke packet arrive at the source

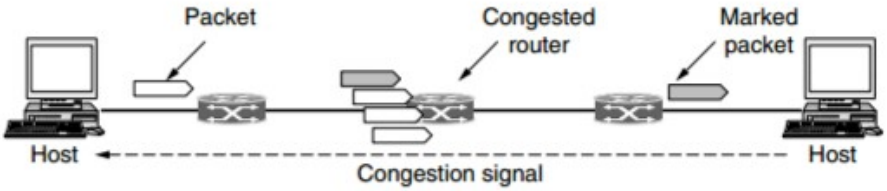


Figure 5-25. Explicit congestion notification

Hop-by-Hop Backpressure:

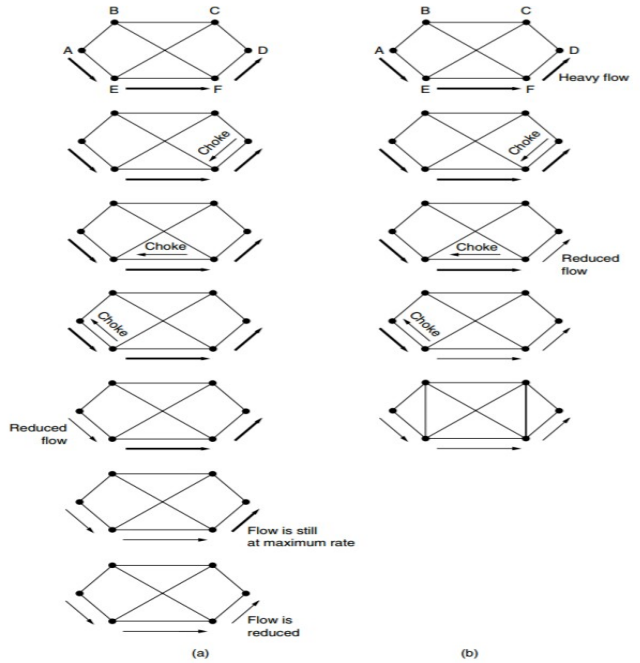


Figure 5-26. (a) A choke packet that affects only the source. (b) A choke packet that affects each hop it passes through.

➤ Load Shedding:

1. When buffers become full, routers simply discard packets.

2. Which packet is chosen to be the victim depends on the application and on the error strategy used in the data link layer.

3. For a file transfer, for, e.g. cannot discard older packets since this will cause a gap in the received data.

4. For real-time voice or video it is probably better to throw away old data and keep new packets. 5. Get the application to mark packets with discard priority.

Quality of service:

- We have to provide good quality of service basing on the demand of applications (and customers), even though we have provided algorithms that reduced congestion and improved network performance.
- Multimedia applications in particular, often need a **minimum throughput** and **maximum latency** to work.
- An easy solution to provide good quality of service is to build a network with enough capacity for whatever traffic will be thrown at it and we call it as **overprovisioning**. The resulting network will carry application traffic without significant loss and, assuming a decent routing scheme, will deliver packets with low latency
- Overprovisioning is expensive to handle. It is based on expected traffic. All bets (expectations) are off if the traffic changes too much.
- Quality of service let a network with less capacity meet application requirements just as well at a lower cost. With quality of service, the network can honor the performance guarantees that it makes even when traffic spikes, at the cost of turning down some requests.

- Four issues must be addressed to ensure quality of service:

1. What applications need from the network?

2. How to regulate the traffic that enters the network.

3. How to reserve resources at routers to guarantee performance.

4. Whether the network can safely accept more traffic.

Several common applications and the stringency of their network requirements are listed below:

Application requirements:

A stream of packets from a source to a destination is called a flow (Clark, 1988). A flow might be all the packets of a connection in a connection-oriented network, or all the packets sent from one process to another process in a connectionless network. The needs of each flow can be characterized by four primary parameters: bandwidth, delay, jitter, and loss.

- Several common applications and the stringency of their network requirements are listed below:

Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

- The variation (i.e., standard deviation) in the delay or packet arrival times is called **jitter.**

To accommodate a variety of applications, networks may support different categories of QoS. An influential example comes from ATM networks, which were once part of a grand vision for networking but have since become a niche technology. They support:

1. Constant bit rate (providing a uniform bandwidth and a uniform delay).
e.g., telephony
2. Real-time variable bit rate (e.g., compressed videoconferencing).
3. Non-real-time variable bit rate (e.g., watching a movie on demand).
4. Available bit rate (e.g., file transfer).

Traffic Shaping:

1. Another method of congestion control is to "shape" the traffic before it enters the network.
2. Traffic shaping controls the rate at which packets are sent (not just how many). Used in ATM and Integrated Services networks.
3. At connection set-up time, the sender and carrier negotiate a traffic pattern (shape).

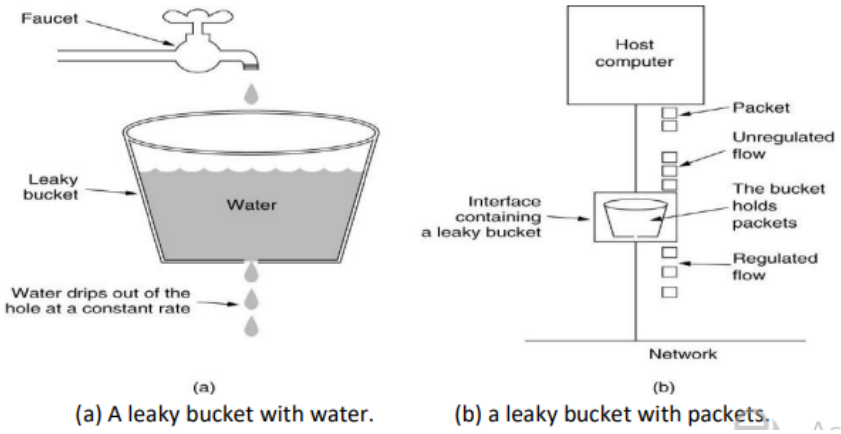
Two traffic shaping algorithms are:

- Leaky Bucket
- Token Bucket

Leaky and Token Buckets:

The Leaky Bucket Algorithm used to control rate in a network. It is implemented as a single server queue with constant service

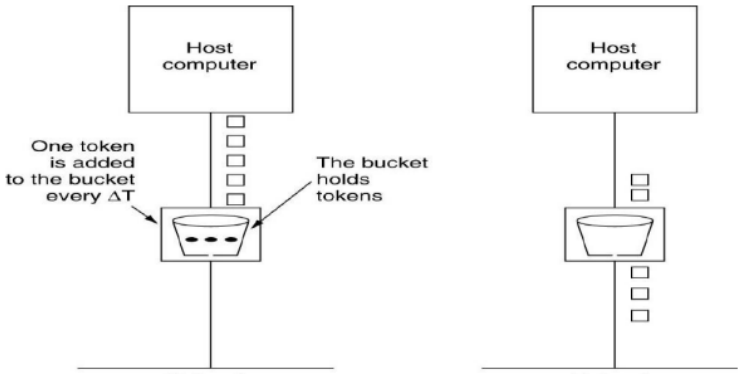
time. If the bucket (buffer) overflows then packets are discarded.



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

1. The leaky bucket enforces a constant output rate (average rate) regardless of the burstiness of the input. Does nothing when input is idle.
2. The host injects one packet per clock tick onto the network. This results in a uniform flow of packets, smoothing out bursts and reducing congestion.
3. When packets are the same size (as in ATM cells), the one packet per tick is okay. For variable length packets though, it is better to allow a fixed number of bytes per tick.
E.g. 1024 bytes per tick will allow one 1024-byte packet or two 512-byte packets or four 256- byte packets on 1 tick

➤ Token Bucket Algorithm:



(a) Before. (b) After.

1. In contrast to the LB, the Token Bucket Algorithm, allows the output rate to vary, depending on the size of the burst.
2. In the TB algorithm, the bucket holds tokens. To transmit a packet, the host must capture and destroy one token.
3. Tokens are generated by a clock at the rate of one token every Δt sec.
4. Idle hosts can capture and save up tokens (up to the max. size of the bucket) in order to send larger bursts later.

➤ Leaky Bucket vs. Token Bucket :

1. LB discards packets; TB does not. TB discards tokens.
2. With TB, a packet can only be transmitted if there are enough tokens to cover its length in bytes.
3. LB sends packets at an average rate. TB allows for large bursts to be sent faster by speeding up the output.
4. TB allows saving up tokens (permissions) to send large bursts. LB does not allow saving.

Packet Scheduling:

To provide a performance guarantee, we must reserve sufficient resources along the route that the packets take through the network. Algorithms that allocate router resources among the packets of a flow and between competing flows are called **packet scheduling algorithms**. Three different kinds of resources can potentially be reserved for different flows:

1. Bandwidth.
2. Buffer space.
3. CPU cycles.

- The first one, bandwidth, is the most obvious. If a flow requires 1 Mbps and the outgoing line has a capacity of 2 Mbps, trying to direct three flows through that line is not going to work. Thus, reserving bandwidth means not oversubscribing any output line.
- A second resource that is buffer space, The purpose of the buffer is to absorb small bursts of traffic as the flows contend with each other. If no buffer is available, the packet has to be discarded since there is no place to put it. For good quality of service, some buffers might be reserved for a specific flow so that flow does not have to compete forbuffers with other flows.
- Finally, CPU cycles may also be a scarce resource. It takes router CPU time to process a packet, so a router can process only a certain number of packets per second. While modern routers are able to process most packets quickly.
- Making sure that the CPU is not overloaded is needed to ensure timely processing of these packets. Packet scheduling algorithms allocate bandwidth and other router resources by determining which of the buffered packets to send on the output line next.
- We already described the most straightforward scheduler when explaining how routers work. Each router buffers packets in a queue for each output line until they can be sent, and they are sent in the same order that they arrived. This algorithm is known as FIFO (First-In First-Out), or equivalently FCFS (First-Come First-Serve),priority scheduling algorithms.
- FIFO scheduling is simple to implement, but it is not suited to providing good quality of service because when there are multiple flows, one flow can easily affect the performance of the other flows.
- **Fair queueing** algorithm :
    1. The essence of this algorithm is that routers have separate queues, one for each flow for a given output line. When the line becomes idle, the router scans the queues round-robin, as shown in Fig. 5-30.
    2. It then takes the first packet on the next queue. In this way, with n hosts competing for the output line, each host gets to send one out of every n packets. It is fair in the sense that all flows get to send packets at the same rate.

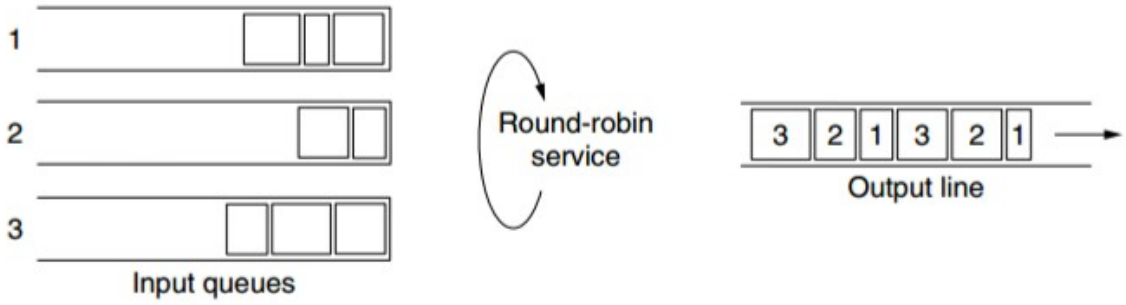


Figure 5-30. Round-robin fair queueing.

Admission Control:

- Quality of Service guarantees are established through the process of admission control. We first saw admission control used to control congestion, which is a performance guarantee.
- The user offers a flow with an accompanying QoS requirement to the network. The network then decides whether to accept or reject the flow based on its capacity and the commitments it has made to other flows.

- Many routing algorithms find the single best path between each source and each destination and send all traffic over the best path. This is called **QoS routing.**
- A simple method is for routers to choose equal-cost paths and to divide the traffic equally or in proportion to the capacity of the outgoing links.
- This may cause some flows to be rejected if there is not enough spare capacity along the best path.
- QoS guarantees for new flows may still be accommodated by choosing a different route for the flow that has excess capacity.
- Because many parties may be involved in the flow negotiation (the sender, the receiver, and all the routers along the path between them), flows must be described accurately in terms of specific parameters that can be negotiated.

Integrated Services:

1. Between 1995 and 1997, IETF put a lot of effort into devising an architecture for streaming multimedia. This work resulted in over two dozen RFCs, starting with RFCs 2205–2212. The generic name for this work is integrated services. It was aimed at both unicast and multicast applications.

RSVP—The Resource reSerVation Protocol:

The main part of the integrated services architecture that is visible to the users of the network is RSVP. It is described in RFCs 2205–2210. This protocol is used for making the reservations; other protocols are used for sending the data.

1. RSVP allows multiple senders to transmit to multiple groups of receivers, permits individual receivers to switch channels freely, and optimizes bandwidth use while at the same time eliminating congestion.

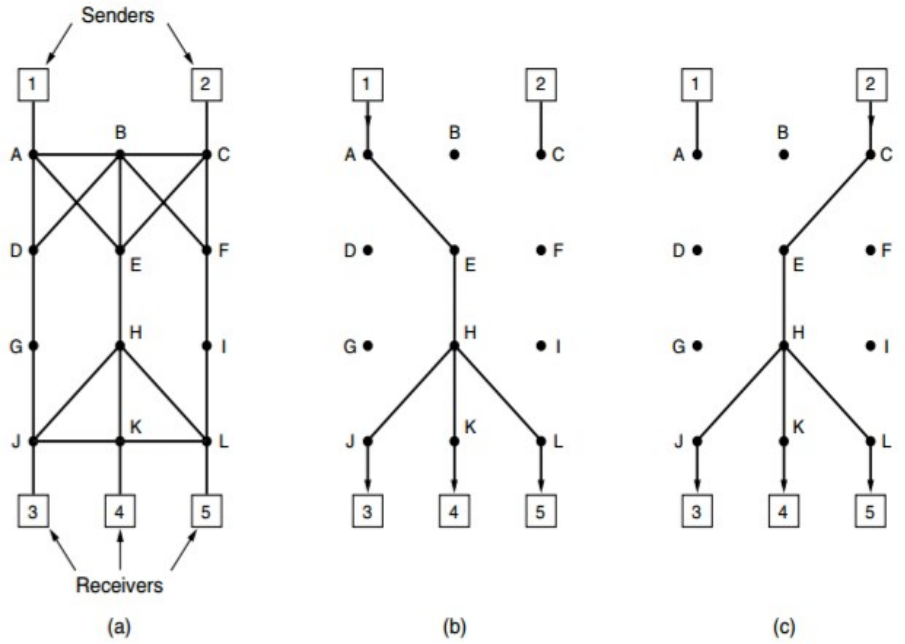


Figure 5-34. (a) A network. (b) The multicast spanning tree for host 1. (c) The multicast spanning tree for host 2.

- As an example, consider the network of Fig. 5-34(a). Hosts 1 and 2 are multicast senders, and hosts 3, 4, and 5 are multicast receivers. In this example, the senders and receivers are disjoint, but in general, the two sets may overlap. The multicast trees for hosts 1 and 2 are shown in Fig. 5-34(b) and Fig. 5-34(c), respectively. To get better reception and eliminate congestion, any of the receivers in a group can send a reservation message up the tree to the sender.
- We saw in the previous section how a weighted fair queueing scheduler can be used to make this reservation. If insufficient bandwidth is available, it reports back failure. By the time the message gets back to the source, bandwidth has been reserved all the way from the sender to the receiver making the reservation request along the spanning tree.
- An example of such a reservation is shown in Fig. 5-35(a). Here host 3 has requested a channel to host 1. Once it has been established, packets can flow from 1 to 3 without congestion.
- Now consider what happens if host 3 next reserves a channel to the other sender, host 2, so the user can watch two

television programs at once.

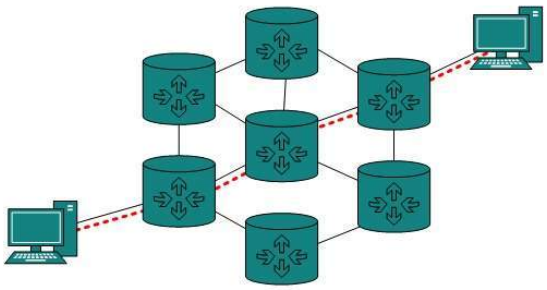
- A second path is reserved, as illustrated in Fig. 5-35(b). Note that two separate channels are needed from host J to router E because two independent streams are being transmitted.
- Finally, in Fig. 5-35(c), host 5 decides to watch the program being transmitted by host I and also makes a reservation. First, dedicated bandwidth is reserved as far as router H.

Internetworking:

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.

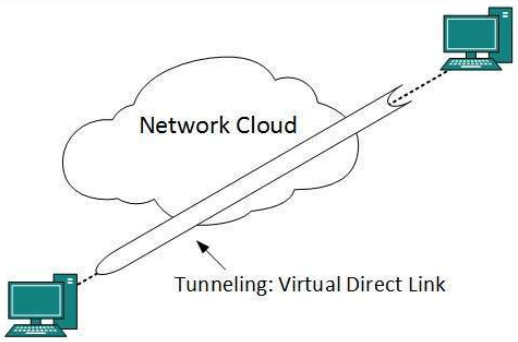


Routing protocols which are used within an organization or administration are called Interior Gateway Protocols or IGP. RIP, OSPF are examples of IGP. Routing between different organizations or administrations may have Exterior Gateway Protocol, and there is only one EGP i.e. Border Gateway Protocol.

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.

➤ Internetwork Routing:

Routing through an internet poses the same basic problem as routing within a single network, but with some added complications.

To start, the networks may internally use different routing algorithms.

For example, one network may use link state routing and another distance vector routing.

Since link state algorithms need to know the topology but distance vector algorithms do not, this difference alone would make it unclear how to find the shortest paths across the internet.

Networks run by different operators lead to bigger problems. First, the operators may have different ideas about what is a good path through the network. One operator may want the route with the least delay, while another may want the

most inexpensive route. This will lead the operators to use different quantities to set the shortest-path costs (e.g., milliseconds of delay vs. monetary cost). The weights will not be comparable across networks, so shortest paths on the internet will not be well

defined.

THE NETWORK LAYER IN THE INTERNET

Top 10 Principles (From Most Important To Least Important) Of Internet Protocol

1. **Make sure it works.** Do not finalize the design or standard until multiple prototypes have successfully communicated with each other. All too often, designers first write a 1000-page standard, get it approved, then discover it is deeply flawed and does not work. Then they write version 1.1 of the standard. This is not the way to go.
2. **Keep it simple.** When in doubt, use the simplest solution. William of Occam stated this principle (Occam's razor) in the 14th century. Put in modern terms: fight features. If a feature is not absolutely essential, leave it out, especially if the same effect can be achieved by combining other features.
3. **Make clear choices.** If there are several ways of doing the same thing, choose one. Having two or more ways to do the same thing is looking for trouble. Standards often have multiple options or modes or parameters because several powerful parties insist that their way is
best. Designers should strongly resist this tendency. Just say no.
4. **Exploit modularity.** This principle leads directly to the idea of having protocol stacks, each of whose layers is independent of all the other ones. In this way, if circumstances require one module or layer to be changed, the other ones will not be affected.
5. **Expect heterogeneity.** Different types of hardware, transmission facilities, and applications will occur on any large network. To handle them, the network design must be simple, general, and flexible.
6. **Avoid static options and parameters.** If parameters are unavoidable (e.g., maximum packet size), it is best to have the sender and receiver negotiate a value rather than defining fixed choices.
7. **Look for a good design; it need not be perfect.** Often, the designers have a good design but it cannot handle some weird special case. Rather than messing up the design, the designers should go with the good design and put the burden of working around it on the people with the strange requirements.
8. **Be strict when sending and tolerant when receiving.** In other words, send only packets that rigorously comply with the standards, but expect incoming packets that may not be fully conformant and try to deal with them.
9. **Think about scalability.** If the system is to handle millions of hosts and billions of users effectively, no centralized databases of any kind are tolerable and load must be spread as evenly as possible over the available resources.
10. **Consider performance and cost.** If a network has poor performance or outrageous costs, nobody will use it.

➤IP Protocol :

Internet Protocols are a set of rules that governs the communication and exchange of data over the internet. Both the sender and receiver should follow the same protocols in order to communicate the data. In order to understand it better, let's take an example of a language. Any language has its own set of vocabulary and grammar which we need to know if we want to communicate in that language. Similarly, over the internet whenever we access a website or exchange some data with another device then these processes are governed by a set of rules called the internet protocols.

➤ IPV4 Header format:

- **IP** stands for **Internet Protocol** and **v4** stands for **Version Four** (IPv4).
- IPv4 was the primary version brought into action for production within the ARPANET in 1983.
- IP version four addresses are 32-bit integers which will be expressed in hexadecimal notation.
- Internet Protocol being a layer-3 protocol (OSI) takes data Segments from layer-4 (Transport) and divides it

into packets. IP packet encapsulates data unit received from above layer and add to its own header information.

Example- 192.0.2.126 could be an IPv4 address.

(IP Encapsulation)

Version (4 bits)	Header Length (4 bits)	Service (8 bits)	Total Length (16bits)
Identification (16 bits)			Flags (3 bits) / Fragment Offset (13 bits)
Time To Live (8 bits)		Protocol (8 bits)	Header Checksum (16 bits)
Source IP Address (32 bits)			
Destination IP Address (32 bits)			
Data (32 bits)			

- **Version** – Version no. of Internet Protocol used (e.g. IPv4).
- **IHL** – Internet Header Length; Length of entire IP header. IP header length (4 bits), which is the number of 32 bit words in the header. The minimum value for this field is 5 and the maximum is 15.
- **DSCP** – Differentiated Services Code Point; this is Type of Service.
- **Total Length** – Length of entire IP Packet (including IP header and IP Payload).
- **Identification** – If IP packet is fragmented during the transmission, all the fragments contain same identification number, to identify original IP packet they belong to.
- **Flags** – As required by the network resources, if IP Packet is too large to handle, these 'flags' tells if they can be fragmented or not. In this 3-bit flag, the MSB is always set to '0'.
- **Fragment Offset** – This offset tells the exact position of the fragment in the original IP Packet.
- **Time to Live** – To avoid looping in the network, every packet is sent with some TTL value set, which tells the network how many routers (hops) this packet can cross. At each hop, its value is decremented by one and when the value reaches zero, the packet is discarded.
- **Protocol** – Tells the Network layer at the destination host, to which Protocol this packet belongs to, i.e. the next level Protocol. For example protocol number of ICMP is 1, TCP is 6 and UDP is 17.
- **Header Checksum** – This field is used to keep checksum value of entire header which is then used to check if

the packet is received error-free.

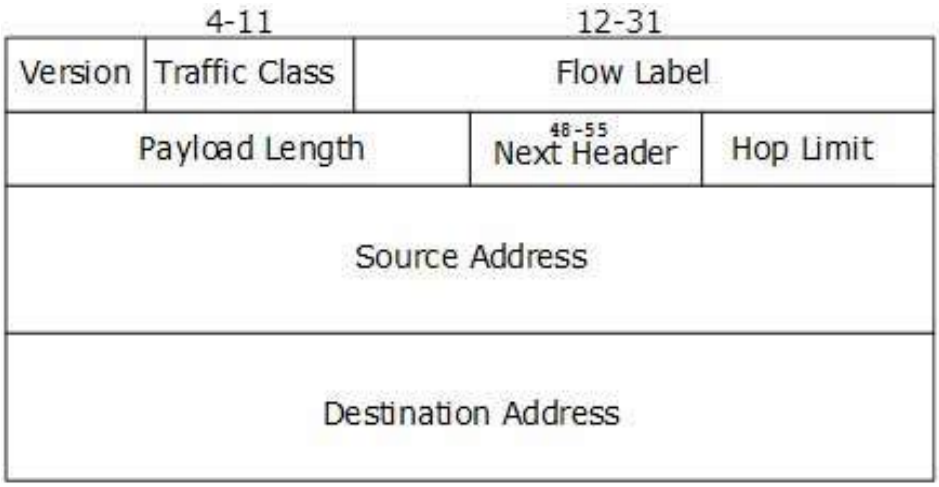
- **Source Address** – 32-bit address of the Sender (or source) of the packet.
- **Destination Address** – 32-bit address of the Receiver (or destination) of the packet.

Hierarchical Addressing Scheme:

- IPv4 uses hierarchical addressing scheme. An IP address, which is 32-bits in length, is divided into two or three parts as depicted –

-
-
- A single IP address can contain information about the network and its sub-network and ultimately the host. This scheme enables the IP Address to be hierarchical where a network can have many sub-networks which in turn can have many hosts.

➤ IPv6 – Header format:



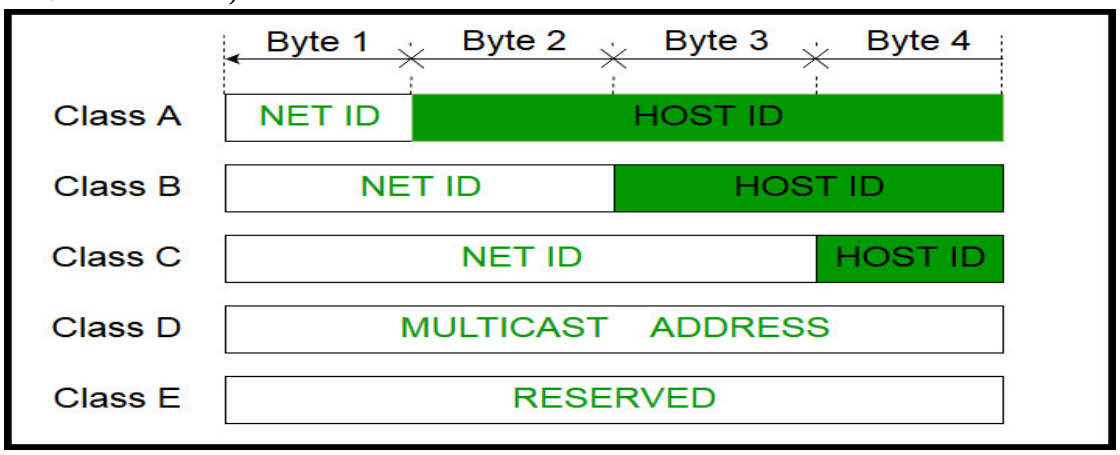
- **Version** (4-bits): It represents the version of Internet Protocol, i.e. 0110.
- **Traffic Class** (8-bits): These 8 bits are divided into two parts. The most significant 6 bits are used for Type of Service to let the Router Known what services should be provided to this packet. The least significant 2 bits are used for Explicit Congestion Notification (ECN).
- **Flow Label** (20-bits): This label is used to maintain the sequential flow of the packets belonging to a communication. The source labels the sequence to help the router identify that a particular packet belongs to a specific flow of information. This field helps avoid re-ordering of data packets. It is designed for

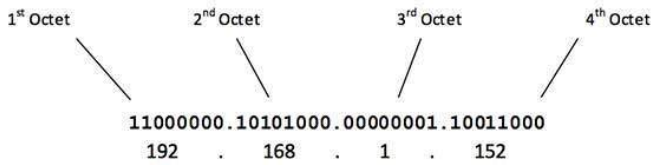
streaming/real-time media.

- **Payload Length** (16-bits): This field is used to tell the routers how much information a particular packet contains in its payload. Payload is composed of Extension Headers and Upper Layer data.
- **Next Header** (8-bits): This field is used to indicate either the type of Extension Header, or if the Extension Header is not present then it indicates the Upper Layer PDU.
- **Hop Limit** (8-bits): This field is used to stop packet to loop in the network infinitely. This is same as TTL in IPv4. The value of Hop Limit field is decremented by 1 as it passes a link (router/hop). When the field reaches 0 the packet is discarded.
- **Source Address** (128-bits): This field indicates the address of originator of the packet.
- **Destination Address** (128-bits): This field provides the address of intended recipient of the packet.

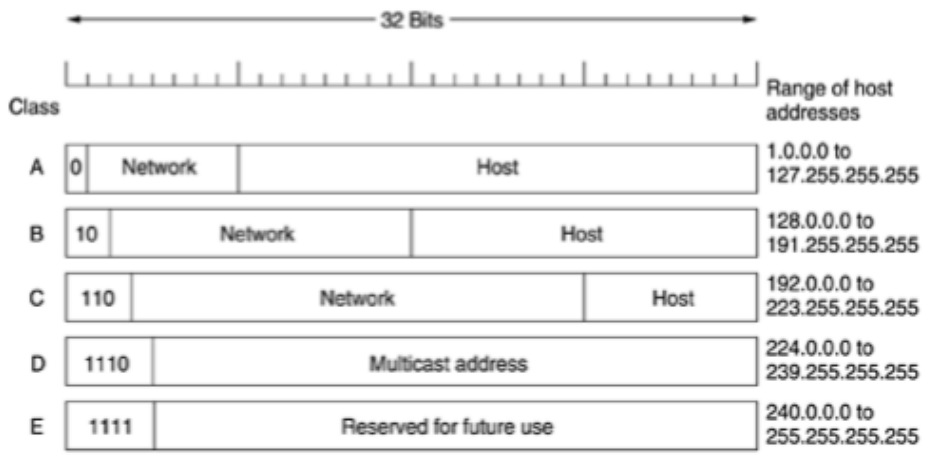
➤ IP Adresses : Classful IP Addreessing: (various types of IP Address)

The first octet referred here is the left most of all. The octets numbered as follows depicting dotted decimal notation of IP Address –

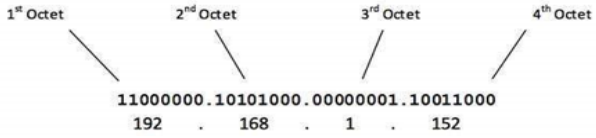




Classful IP Addreessing: (various types of IP Address)



The first octet referred here is the left most of all. The octets numbered as follows depicting dotted decimal notation of IP Address –



Class A Address

The first bit of the first octet is always set to 0 (zero). Thus the first octet ranges from 1 – 127, i.e.

00000001 – 01111111
1 – 127

Class A addresses only include IP starting from 1.x.x.x to 126.x.x.x only. The IP range 127.x.x.x is reserved for loopback IP addresses.
The default subnet mask for Class A IP address is 255.0.0.0 which implies that Class A addressing can have 126 networks (2^7-2) and 16777214 hosts (2^{24}-2).
Class A IP address format is thus:

0000 00000. 0000 0000. 0000 0000 . 0000 0000
01111 1111 . 0000 0000 . 0000 0000 . 0000 0000

Class B Address

An IP address which belongs to class B has the first two bits in the first octet set to 10, i.e.

10000000 – 10111111
128 – 191

Class B IP Addresses range from **128.0.x.x to 191.255.x.x.** The default subnet mask for Class B is 255.255.x.x.
Class B has 16384 (2^{14}) Network addresses and 65534 (2^{16}-2) Host addresses.
Class B IP address format is:

10**NNNNNN.NNNNNNNN**.HHHHHHHH.HHHHHHHH

1000 0000. 0000 0000. 0000 0000 . 0000 0000
10 11 1111 . 0000 0000 . 0000 0000 . 0000 0000

Class C Address

The first octet of Class C IP address has its first 3 bits set to 110, that is –

11000000 – 11011111
192 – 223

11 0 0 00 00. 0000 0000. 0000 0000 . 0000 0000
11 0 1 11 11 . 0000 0000 . 0000 0000 . 0000 0000

Class C IP addresses range from 192.0.0.x to 223.255.255.x. The default subnet mask for Class C is 255.255.255.x.
Class C gives 2097152 (2^{21}) Network addresses and 254 (2^{8}-2) Host addresses.
Class C IP address format is:

110**NNNNN.NNNNNNNN.NNNNNNNN**.HHHHHHHH

Class D Address:

Very first four bits of the first octet in Class D IP addresses are set to 1110, giving a range of –

11100000 – 11101111
224 – 239

Class D has IP address range from 224.0.0.0 to 239.255.255.255. Class D is reserved for Multicasting. In multicasting data is not destined for a particular host, that is why there is no need to extract host address from the IP address, and Class D does not have any subnet mask.

Class E Address:

This IP Class is reserved for experimental purposes only for R&D or Study. IP addresses in this class ranges from 240.0.0.0 to 255.255.255.254. Like Class D, this class too is not equipped with any subnet mask.