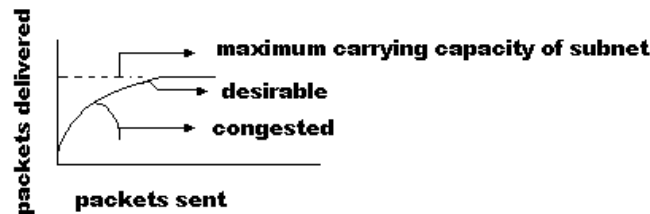


**Congestion:** When too many packets are sent to a subnet more than its capacity, the Situation that arises is called **congestion**.



### **Reasons for Congestion:**

1. If input packets coming from 3 or 4 lines, requires only one particular output line.
2. If routers are supplied with infinite amount of memory, packets take longtime to reach to the front of queue where duplicates are generated as they are timed out.
3. Slow processors cause congestion.
4. Low bandwidth lines also cause congestion.
5. Congestion feeds upon itself and cause congestion.

### **Congestion Control Algorithms:**

These algorithms control congestion. These are mainly divided into two groups:

1. Open Loop Solutions.
2. Closed Loop Solutions.

**Open Loop Solutions** attempt to solve the problems by good design to make sure it does not occur in the first place. Once the system is up and running, mid course corrections are not made.

**Closed Loop Solutions** are based on the concepts of a feedback loop. It has 3 parts.

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

These closed loop algorithms are further divided into two categories:

- **Implicit feedback:** The source reduces the congestion existence by making local observations.
- **Explicit feed back:** Packets are sent back from the point of congestion to warn source

**Open Loop Solutions:**

- Congestion Prevention Policies
- Traffic Shaping
- Flow Specifications

**1. Congestion prevention policies:**

Congestion is prevented using appropriate policies at various levels.

Layer	Policies
<b>Transport</b>	<ol style="list-style-type: none"> <li>1. Retransmission policy</li> <li>2. Out-of-order caching policy</li> <li>3. Acknowledgement policy</li> <li>4. Flow control policy</li> <li>5. Timeout Determination</li> </ol>
<b>Network</b>	<ol style="list-style-type: none"> <li>1. Virtual circuits versus data gram inside the subnet</li> <li>2. Packet queuing service policy</li> <li>3. Packet discard policy</li> <li>4. Routing algorithm</li> <li>5. Packet lifetime Management</li> </ol>
<b>Data Link</b>	<ol style="list-style-type: none"> <li>1. Retransmission policy</li> <li>2. Out-of-order catching policy</li> <li>3. Acknowledgement policy</li> <li>4. Flow control policy</li> </ol>

**Retransmission policy:** Deals with how fast a sender times out and what it transmits upon time out.

**Out-of –order Catching policy:** If receivers routinely discard all out-of-order packets, (packets arrived without order) they have to be retransmitted.

**Acknowledgement policy:** If each packet is acknowledged immediately, acknowledged packets generate extra traffic. This policy deals with piggybacking.

**Flow Control policy:** A tight flow control scheme (ex: a small window) reduces the data rate and thus helps fight congestion.

**Timeout Determination:** It is harder as transit time across the network is less predictable than transit time over a wire between two routers.

**Virtual Circuits vs Data grams:** This affects congestion as many algorithms work only with virtual circuits.

**Packets queuing and Service policy:** Relates to whether routers have one queue per input line, one queue per output line or both.

**Packet Discard policy:** Tells which packet is dropped when there is no space.

**Routing Algorithm:** With this, Traffic is spreaded over all the lines.

**Packet lifetime management:** Deals with how long a packet may live before being discarded.

## **2. Traffic Shaping:**

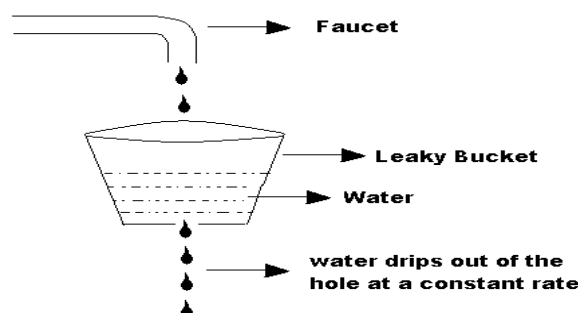
It is the process of forcing the packets to be transmitted at a more predictable rate. This approach is widely used in ATM Networks to manage congestion. When a virtual circuit is set up, the user and the subnet agree on a certain traffic pattern for that circuit. Monitoring a traffic flow based on agreement made is called "**Traffic Policing**".

**Traffic shaping** can be implemented with any of the two techniques:

- **Leaky Bucket Algorithm**
- **Token Bucket Algorithm**

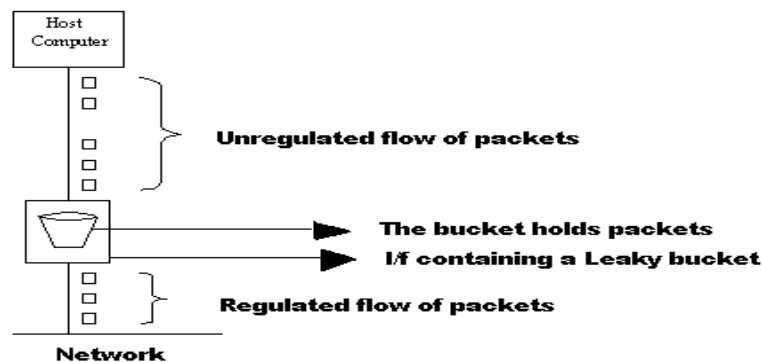
### **The Leaky Bucket Algorithm:**

Imagine a bucket with a small hole in the bottom. No matter, at what rate water enters the bucket, the outflow is at a constant rate, ' $p$ ', when there is any water in the bucket and ' $r$ ', when the bucket is empty. Also, once the bucket is full, any additional water entering it spills over the sides and is lost. The same idea can be applied to packets.



Conceptually, each host is connected to the network by an interface containing a leaky bucket, i.e., a finite internal queue. If a packet arrives at the queue when it is full, it is discarded. In other

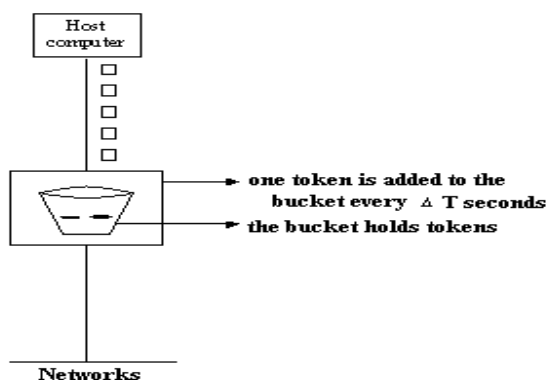
words, if one or more processes within the host try to send a packet when the maximum number are already queued, the new packet is unceremoniously discarded. This arrangement can be built into the h/w interface or simulated by the host operating system. It was first proposed by **TURNER** and is called the “**LEAKY BUCKET ALGORITHM**”.



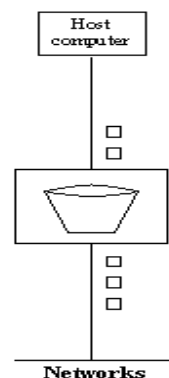
The host is allowed to put one packet per clock tick onto the network, which turns an uneven flow of packets from the user processes inside the host into an even flow of packets onto the network, smoothing out bursts and greatly reducing the chances of congestion.

**The Token Bucket Algorithm:** The algorithm that allows the output to speedup when large bursts arrive and one that never loses data is the **TOKEN BUCKET ALGORITHM**. In this algorithm, the leaky bucket holds tokens, generated by a clock at the rate of one token every  $\Delta T$  sec. This algorithm allows to save up permission by hosts, up to the maximum size of the bucket, ‘n’ i.e., bursts of up to ‘n’ packets can be sent at once, allowing some burstiness in output stream and giving faster response to sudden bursts of input.

**Before Transmission**



**After Transmission**



In the above circuit, we see a bucket holding 3 tokens, with 5 packets waiting to be transmitted. For a packet to be transmitted, it must capture and destroy one token. In the above example, 3 out of 5 packets have gotten through by capturing the 3 tokens in the bucket, but the other 2 are stuck waiting for 2 more tokens to be generated. The implementation of the token bucket algorithm is just a variable that counts tokens. The counter is incremented by 1, every  $\Delta T$  and decremented by 1, when a packet is sent. When the counter hits '0', no packets may be sent.

**The major advantage of the token bucket algorithm is that it throws away tokens instead of packets, when the bucket fills up.**

### **(3). Flow Specifications:**

The agreement made to specify the traffic pattern in a precise way is called a Flow-Specification. It consists of data structure that describes both the pattern of the injected traffic and the quality of service desired by the applications. This can be applied to both virtual circuits and datagrams. Before a connection is established, the source gives the flow specification to the subnet for approval, which can be accepted, rejected or given a counter proposal. Once the sender and subnet have struck a deal, the sender can ask the receiver if it, too, agrees.

#### **An example flow specification:**

Injected Traffic Pattern	Desired Service
<ul style="list-style-type: none"> <li>• Maximum Packet Size(bytes)</li> <li>• Token Bucket Rate (bytes/sec)</li> <li>• Token Bucket Size(bytes)</li> <li>• Maximum Transmission Rate(bytes/sec)</li> </ul>	<ul style="list-style-type: none"> <li>➤ Loss Sensitivity (bytes)</li> <li>➤ Loss Interval (<math>\mu\text{sec}</math>)</li> <li>➤ Burst Loss Sensitivity(packets)</li> <li>➤ Minimum Delay Noticed(<math>\mu\text{sec}</math>)</li> <li>➤ Maximum Delay Variation(<math>\mu\text{sec}</math>)</li> <li>➤ Quantity Of Guarantee</li> </ul>

**Maximum packet size:** Tells how big packets may be.

**Token bucket rate:** Tells how many bytes are put into the token bucket per second

**Token bucket size:** Tells how big the bucket is.

**Maximum transmission rate:** It is top rate, host is capable of producing, under any conditions.

**Burst loss sensitivity:** Tells how many consecutive lost packets can be tolerated.

**Loss sensitivity** } Represent the numerical & denominator of a fraction giving the maximum  
**Loss interval** } acceptable loss rate.

**Minimum delay noticed**: Tells how long a packet can be delayed without noticed by application

**Maximum delay variation**: Tries to quantify the fact that some applications are not sensitive to the actual delay but one highly sensitive to jitter.

**Quality of guarantee**: Indicates whether or not the application really needs it.

**Minimum delay noticed**: Tells how long a packet can be delayed without noticed by application

**Maximum delay variation**: Tries to quantify the fact that some applications are not sensitive to the actual delay but one highly sensitive to jitter.

**Quality of guarantee**: Indicates whether or not the application really needs it.

### **Closed loop algorithms:**

- Congestion control in virtual circuit subnets
- Choke packets
- Load Shedding
- Jitter Control
- Congestion control for Multicasting

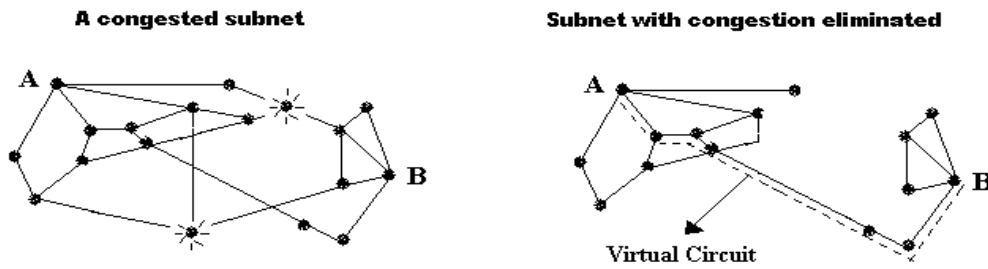
### **1. Congestion control in virtual circuit subnets:**

Congestion can be controlled using any of the following techniques:

- Admission control
- Careful Routing
- Agreement Negotiation

**Admission control**: The idea is that, once congestion has been signaled, no more virtual circuits are setup until the problem has gone away. Thus, attempts to setup new transport layer connections fail.

**Careful routing**: New virtual circuits are allowed, but with careful routing around problem areas



**Agreement Negotiation:** An agreement is to be negotiated between the host and subnet when a virtual circuit is set up. This agreement normally specifies the volume and shape of the traffic, quality of service required and other parameters. To keep its part of agreement, the subnet will typically reserve resources along the path when the circuit is set up, and thus avoids congestion.

## **2. Choke packets:**

This is an approach that can be used in both virtual circuit and data gram subnets. Each router can easily monitor the utilization of its output lines and other resources using the formula....

$$u_{\text{new}} = au_{\text{old}} + (1-a)f$$

Where  $u \rightarrow$  a variable that reflects the recent utilization of a line. Value lies between 0.0 and 1.0

$a \rightarrow$  constant that determines how fast the router forgets recent history.

$f \rightarrow$  a sample of instantaneous line utilization (either 0 or 1)

## **Working:**

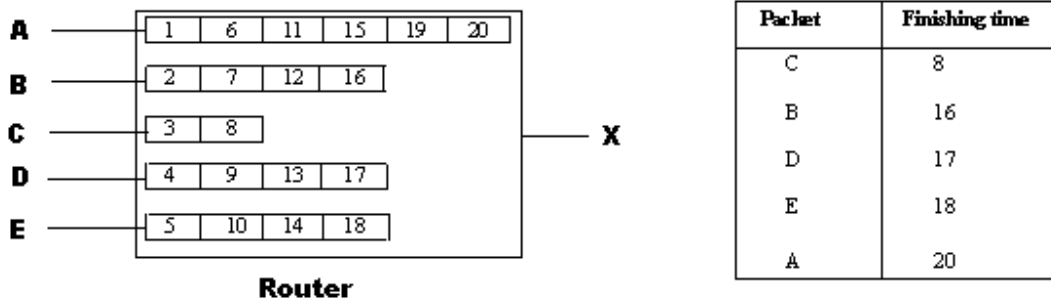
Whenever 'u' moves above the threshold, the output line enters a "**Warning**" state. Each newly arriving packet is checked to see if its output line is in warning state. If so, the router sends a **choke packet** back to the source host, giving it the destination found in the packet. The original packet is tagged so that it will not generate any more choke packets further along the path and is then forwarded in the usual way. When the source host gets the choke packet, It is required to reduce the traffic sent to the specified destination by 'x' percent, i.e., the host should ignore choke packets referring to that destination for a fixed time interval. After that period has expired, the host listens for more choke packets for another interval. If no choke packets arrived during the listening period, the host may increase the flow again.

**Disadvantage:** The honest host gets an ever-smaller share of the bandwidth than it had before.

To get around the problem of choke packets, Nagle proposed “**FAIR QUEUEING ALGORITHM**”. The essence of this algorithm is that routers have multiple queues for each o/p line, one for each source. When a line becomes idle, the router scans the queues round robin, taking the first packet on the next queue. In this way, with ‘n’ hosts competing for a given o/p line, each host gets to send one out of every ‘n’ packets.

**Drawback:** With this algorithm, more bandwidth is given to hosts that use large packets than to hosts that use small packets.

To get around the problem of the algorithm proposed by Nagle, Demers suggested an improvement in which the around robin is done in such away as to simulate a byte-by-byte round robin, instead of a packet-by-packet round robin. It scans the queues repeatedly, byte-for byte, until it finds the tick on which each packet will be finished. The packets are then sorted in the order of their finishing and sent in that order.



In the above example,

- At clock tick 1 - The first byte of packet in the queue on line ‘A’ is sent.
- At clock tick 2 - The first byte of packet in the queue is sent on line ‘B’.
- 
- 
- 
- At clock tick 8 - ‘C’ finishes its first packet and then similarly B, D, E and A finishes after 16 17 ,18 and 20 clock ticks respectively.

**Problem:** It gives the all hosts the same priority.



To overcome this problem of FAIR QUEUEING algorithm, **WEIGHTED FAIR QUEUEING** algorithm is widely used in which the required hosts can be assigned a higher priority or bandwidth so that they can be given 2 or more bytes per tick.

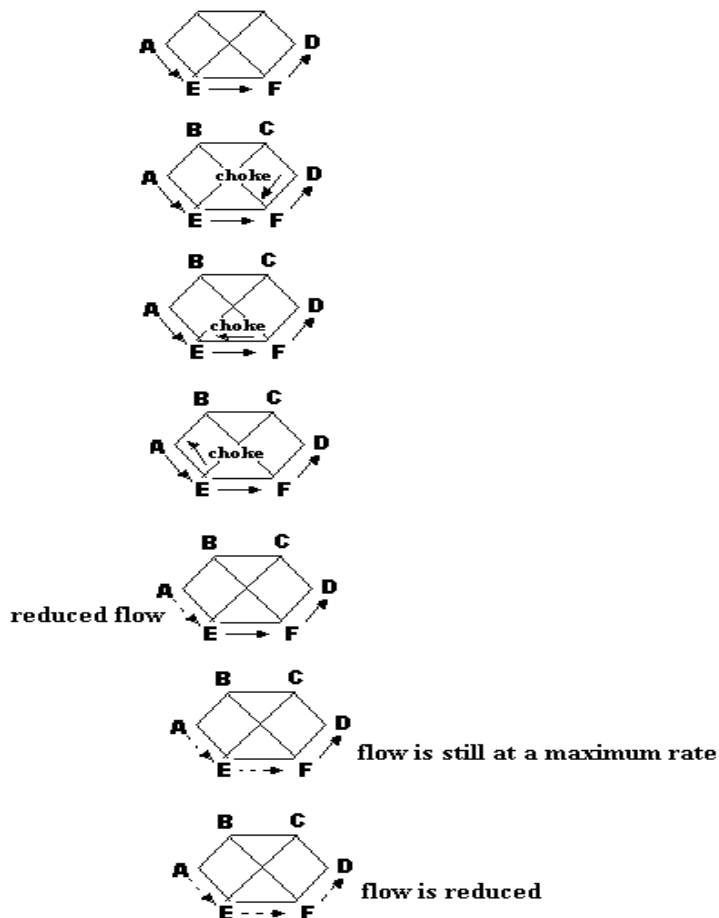
### Hop-by-Hop Choke Packets :

At high speeds and over long distance, sending a choke packet to the source host does not work well because the reaction is slow.

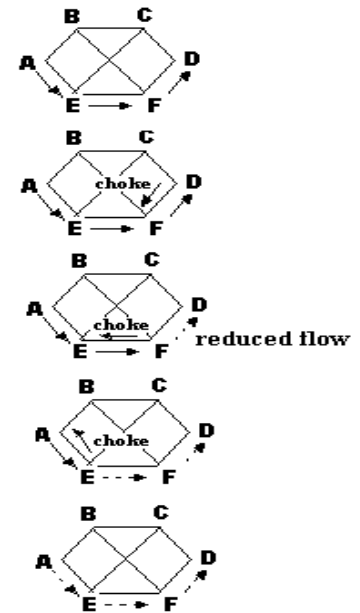
**Eg:** consider fig(i):

Host 'A' sending traffic to host 'B' located at a very long distance from 'A'. The choke packets are released at the time of congestion and as the 2 hosts are far situated from each other, It takes a maximum delay for choke packets to reach the host 'A' and reaction is similar.

#### (i). A Choke packet that effects source



#### (ii). A Choke packet that effects each hop it passes through



An alternative approach to reduce the delay is to have the choke packets take effect at every hop it passes through, as shown in sequence of fig(ii) . Here, as soon as choke packet reaches F, 'F' is required to reduce the flow to 'D'. Doing so will require 'F' to devote more buffers to the flow ,since the source is still sending away at full blast , but it gives 'D' immediate relief . In the next step, packet reaches E , which tells E to reduce the flow to F .This action puts a greater demand on E's buffers but gives 'F' immediate relief . Finally, the choke packet reaches A and the flow genuinely slows down.

### **3. Load Shedding :**

It is a fancy way of saying that when routers are being loaded by packets that they cannot handle, they just throw them away. Which packets to discard depend on the application running.

If a new packet is more important than the old one,  
old packet is removed and this process is called '**MILK**'.

If an old packet is more important than the new one,  
new packet is removed and this process is called '**WINE**'.

### **(4). Jitter Control :**

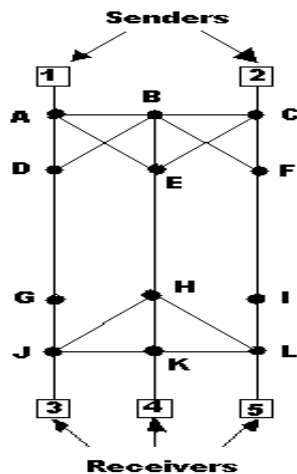
This is based on calculating the average amount of congestion. The jitter can be bounded by computing the expected transit time for each loop along the path. When a packet arrives at a router, the router checks to see how much the packet is behind or ahead of its schedule. This information is stored in the packet and updated at each hop. If the packet is ahead of schedule, it is held just long enough to get it back on schedule. If it is behind schedule, the router tries to get it out the door quickly.

In fact, the algorithm for determining which of several packets competing for an o/p line should go next can always choose the packet furthest behind its schedule. In this way, packets that are ahead of schedule get slowed down and packets that are behind its schedule get speeded up, in both cases reducing the amount of Jitter.

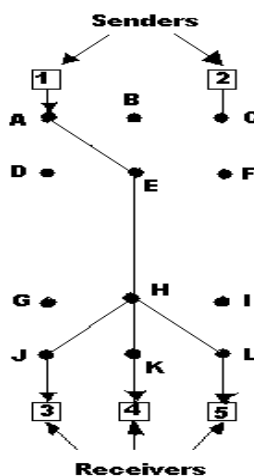
**(5). Congestion Control For Multicasting:**

A way of managing multicast flows from multiple sources to multiple destination is to use the **RSVP (Resource reSerVation Protocol)**, which optimizes bandwidth use while at the same time eliminates congestion. The protocol uses multicast routing using spanning trees. Each group is assigned a group address. To send to a group, a sender puts group address in its packets. The algorithm then builds a spanning tree covering all group members.

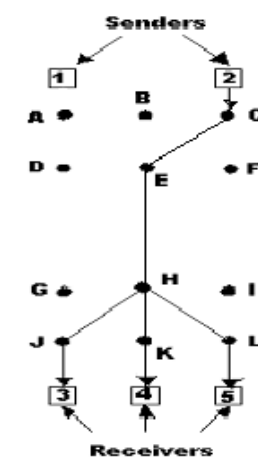
**Example: - A network**



**multi-cast spanning  
tree for host-1**



**multi-cast spanning  
tree for host-2**



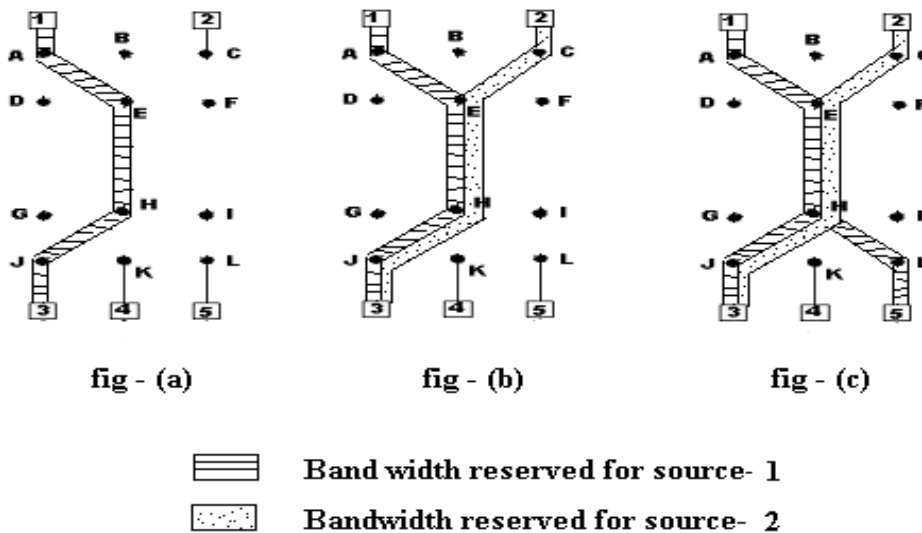
To get better reception and eliminate congestion, any of the receivers in a group can send a reservation message up the tree to the sender. The message is propagated using Reverse path forwarding algorithm. At each hop, the router notes the reservation and reserves the necessary bandwidth. If insufficient 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.

**Example:**

→ Suppose host 3 has requested a channel to host 1. Once it has been established, packets can flow from 1 to 3 without congestion (as in **fig-a**).

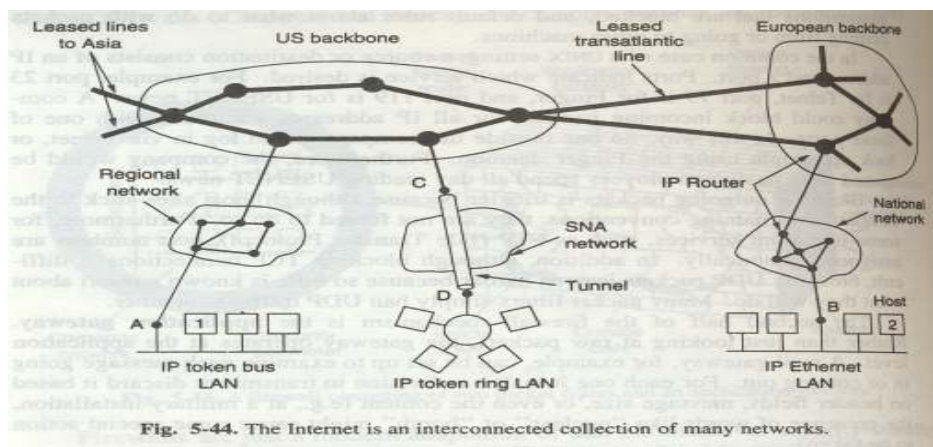
→ Now, consider what happens if host 3 next reserves a channel to other sender, host 2. A second path is reserved. (as in **fig-b**).

→ Finally if host 5 decides to watch program being transmitted by host 1, it makes a reservation. First, dedicated bandwidth is reserved to reach host 1 has already been reserved. So, it does not have to reserve any more (as in **fig-c**) .



## The Network Layer in the Internet

At the network layer, the Internet can be viewed as a collection of sub networks or Autonomous systems, that are connected together. Several backbones exist which are constructed from high bandwidth lines and fast routers. LANs at many Universities, Companies and Internet Service. Providers are attached to Regional Networks, which in turn, are attached to the backbones.

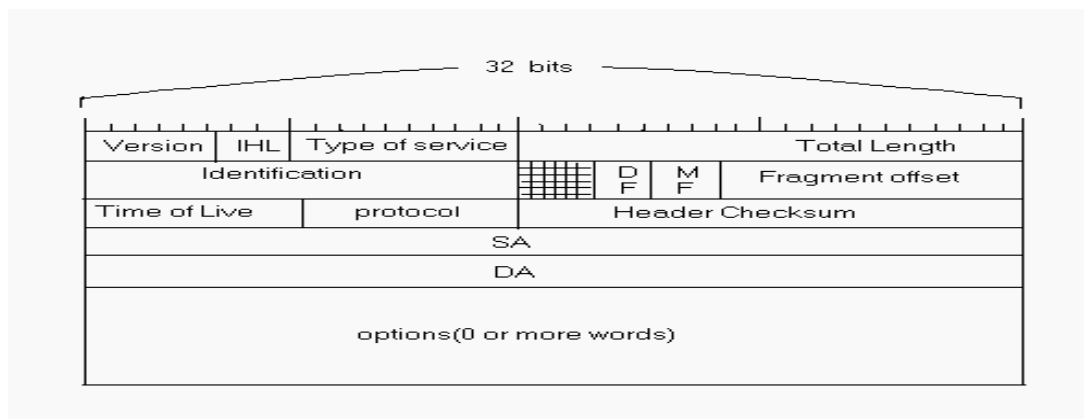


The glue that holds the Internet together is the network layer protocol, IP. The job IP is to provide a best-efforts way to transport datagrams from source to destination, without regard to whether or not these machines are on the same network or there are other networks or there are other networks in between them.

### **Communication in Internet:**

The transport layer takes data streams and breaks them up into datagrams, which are transmitted through the Internet, possibly fragmented into smaller units as it goes. When all pieces finally get to the destination, they are reassembled by the network layer into the original datagram, which is then handed to transport layer.

**The IP Protocol:** The IP protocol (or the IP datagram ) consists of 2 parts: -



**Header part :** The header has a 20-bytes fixed part and a variable length optional part. It is transmitted in big Indian order i.e., from left to right.

**Version:** Keeps track of which version of the protocol the datagram belongs to.

**IHL:** Tells how long the header is, in 32-bit words. The min and max values are 5 and 15 respectively, which are a multiple of 4.

**Type of service:** Allows the host to tell the subnet what kind of service it wants.

The field itself contains:

- 8-bit precedence field, where precedence field is a priority from 0-7.
- 8-flags D, T, R (Delay, Throughput, and Reliability) which is most cared parameter set.
- 2-bits, unused.

**Total Length:** The total length of both header and data maximum length is 65,535 bytes.

**Identification:** Allows the destination host to determine to which datagram a newly arrived fragment belongs.

**DF( Don't Fragment ):** It is an order to routine not to fragment the datagram because the destination is incapable of putting the pieces back together again.

**MF( More Fragments ):** All fragments except the last one have this bit set.

**Fragment Offset:** Tells where in the current datagram this fragment belongs.

**Time to live:** It is a counter used to limit packet lifetimes. It counts time in seconds, allowing a maximum lifetime of 255sec. It is decremented at each hop till it reaches zero and then discarded.

**Header Checksum:** Verifies the header only.

**Protocol:** Tells the network layer to which transport process, the datagram is to be given.

Eg :- TCP, UDP .... etc

**SA:** Indicates the network number and host number from which datagram has come from.

**DA:** Destination address and it indicates both the n/w number and host number to which destination, the datagram is to be delivered.

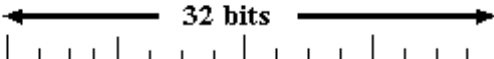
**Options:** This was designed to provide an escape to allow subsequent versions of protocol to include information not present in the original design, to permit experimenters to tryout new ideas and to avoid allocating header bits to information that is rarely needed.

Option	Description
Security	Specifies how secret the datagram is.
Strict Source Routing	Gives the complete path to be followed.
Loose Source Routing	Gives a list of routers not to be missed.
Record Route	Makes each router append its IP address.
Time Stamp	Makes each router append its address and time stamp.

**IP Addresses :** IP addresses = Network Number + Host Number. All are 32-bits long.

Network numbers are assigned by NIC (Network Information Centre) and are usually written in **Dotted Decimal Notation**, in which each of 4 bytes is written in decimal, from 0 to 255.

**Different address formats:**

class											Range of host addresses
A	0	Network				Host					1.0.0.0 to 127.255.255.255
B	1 0	Network				Host					128.0.0.0 to 191.255.255.255
C	1 1 0	Network				Host					192.0.0.0 to 223.255.255.255
D	1 1 1 0	Multicast addresses									224.0.0.0 to 239.255.255.255
E	1 1 1 1 0	Reserved for future use									240.0.0.0 to 247.255.255.255

Class	No. of networks allowed	No. of hosts allowed in each N/W
A	126	16 million
B	16,382	64,000
C	2 million	254

- The lowest IP address is 0.0.0.0.
- The highest IP address is 255.255.255.255.
- The value '0' means this n/w or this host.
- The value '-1' means Broadcast messages to all hosts on the indicated n/w.

**Special IP addresses :**

00000000000000000000000000000000	This host
00.....00      Host	A host on this network
11111111111111111111111111111111	Broadcast on local network
Network    1111.....1111	Broadcast on distant network
127      (Any thing)	Loopback

**Problems :**

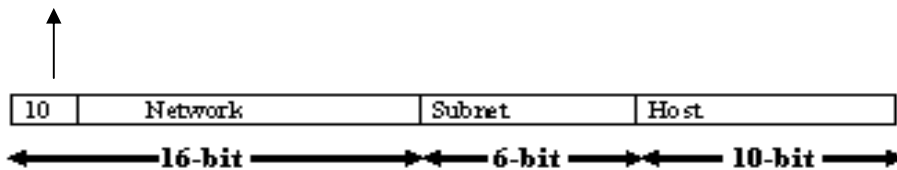
- As time goes on, any class network may acquire more than the permitted no. of hosts which require another class network of same type with a separate IP address.
- As the no. of distinct local n/w s grows, managing them can become a serious headache.

**To overcome all these problems.....**

A network is allowed to split into several parts for internal use but still act like a single network to the outside world. In the internal literature, these parts are called **SUBNETS**.

**Eg :** A company started up with a class B address and had grown as time passed by which require a second LAN. Then, 16-bit host number is splitted up into a 6-bit subnet number and a 10-bit host number. This split allows 62 LANs (0 and –1 are reserved), each with up to 1022 hosts.

**Subnet mask**



In this example, the subnet might use IP address starting at 130.50.4.1, the second subnet might start at 130.50.8.1, and so on.

### **Subnet Working:**

Each router has a table listing some number of (network, 0) IP addresses and some number of (this- n/w, host) IP addresses.

(network, 0)	–	Tells how to get to distant networks
(this – n/w, host)	–	Tells how to get to local hosts

Associated with each table is the network interface to use to reach the destination, and certain other information. When an IP Packet arrives, its destination addresses is looked up in the routing table. If the packet is for a distant network, it is forwarded to the next router on interface given in the table. If it is a local host, it is sent directly to the destination. If the network is not present, the packet is forwarded to default router.

When sub netting is introduced, the routing tables are changed, adding entries of the form (this – n/w, subnet, 0) and (this – n/w, this subnet, host) Thus, a router on a subnet ‘k’ knows how to get all other subnets and also how to get to all the hosts on subnet ‘k’. Each router performs a Boolean AND with network’s subnet mask to get rid of host number and looks up the resulting address in its tables.



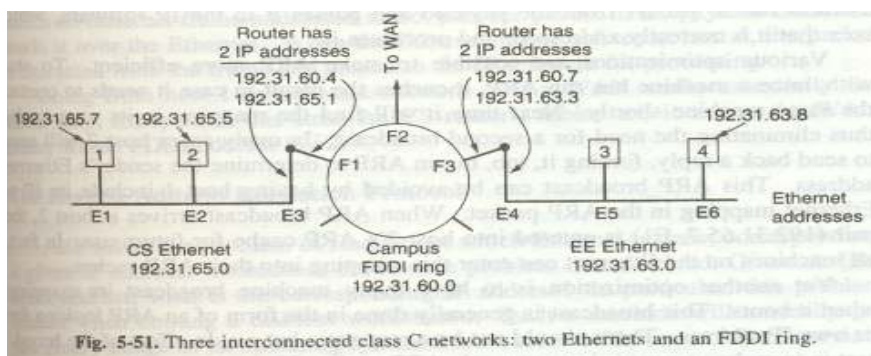
**Internet Control protocols:**

- **ICMP** (Internet Control Message Protocol)
- **ARP** (Address Resolution Protocol)
- **RARP** (Reverse Address Resolution Protocol)
- **BOOTP** (Boot Strap Protocol)

**ICMP:** When some thing unexpected occurs in the internet, the event is reported by **ICMP**, which is used to test the internet. Each ICMP message type is encapsulated in an IP packet.

Message type	Description
Destination Unreachable	Used when the subnet or a router cannot locate the destination or a packet with DF bit can't be delivered
Time exceeded	Is sent when a packet is dropped due to its counter reaching zero(time to live=0)
Source Quench	Used to throttle hosts that were sending too many packets, by which hosts are expected to slow down.
Echo Request Echo Reply	To see if a given destination is reachable and alive. By which the destination machine responds.
Time Stamp Request Time Stamp Reply	
Parameter Problem	Indicates that a illegal value has been detected in header field
Redirect	Used when a router notices that a packet seen to be routed wrong

**ARP:** Used when an IP address is given and corresponding Ethernet address is to be found out. IP addresses get mapped onto data link layer addresses in the following way:-



Here, we have 2 Ethernets, one in dept-1 with IP address 192.31.65.0 and the other in dept-2 with IP address 192.31.63.0 in a university, which are connected by a campus FDDI ring with IP address 192.31.60.0. Each machine on the Ethernet has a unique Ethernet address, labeled  $E_1$  through  $E_6$  and each machine on the FDDI ring has an FDDI address, labeled  $F_1$  through  $F_3$ .

Let us assume the data transfer from user on host1 to user on host2, in which the sender knows the name of the intended receiver, say, “**Mary @ eagle.cs.uni.edu**”:

1. Find the IP address for host-2, **eagle.cs.uni.edu**, which is performed by DNS
2. The IP address for host-2 is returned (192.31.65.5)
3. The upper layer on host-1 now builds a packet with 192.31.65.5 in **Destination Address** field and gives it to the IP software to transmit.
4. The IP s/w finds that the destination is on its own n/w but it doesn't find the destinations Ethernet address. To do so, host-1 puts a broadcast packet onto the Ethernet asking “**Who Owns IP address 192.31.65.5?**”
5. The broadcast will arrive at every machine on Ethernet 192.31.65.0 and each one will check its IP address, by which host-2 will respond with its Ethernet address( $E_2$ )  
( The protocol for asking this question and getting the reply is called **Address Resolution Protocol**, which is defined in RFC826)
6. Now, the IP s/w on host-1 builds an Ethernet frame addressed to  $E_2$ , puts the IP packet in the payload field, and dumps it onto the Ethernet, which is detected and recognized by host2 as a frame for itself and it causes an interrupt.
7. The Ethernet driver extracts the IP packet from the payload and passes it to the IP s/w, which sees that it is correctly addressed, and processes it.

Let us assume the data transfer from user on host1 to user on host6, in which the sender knows the name of the intended receiver, say, “**Mary @ eagle.cs.uni.edu**”

1. ARP in this case, will fail, as host4 will not see the broadcast(Routers don't forward Ethernet level broadcast) 2 solution are possible to deal with this task:-
  - The dept-1 Router could be configured to respond to ARP requests for n/w 192.31.63.0, in which host-1 will make an ARP cache entry of (192.31.63.8,  $E_3$ ) and happily send all traffic for host-4 to the local router. This solution is called “**PROXY ARP**”

- To have host-1 immediately see that the destination is on a remote n/w and just send all such traffic to default Ethernet address that handles all remote traffic, in this case  $E_3$
- 2. i.e., host-1 packs the IP packet into the payload field of an Ethernet addressed to  $E_3$ .
- 3. when the dept-1 router gets the Ethernet frame, it removes the IP packet from the payload field and looks up the IP address in its routing tables
- 4. It discovers that packets for n/w 192.31.63.0 are supposed to go to routes 192.31.60.7. if the FDDI address is not known in prior, ARP technique is used to find the ring address as  $F_3$  by which the packet is inserted into the payload field of an FDDI frame addressed to  $F_3$  and puts it on the ring
- 5. At the dept-2 router, the FDDI driver removes the packet from payload field and gives it to IP software, which sees that it needs to send the packet to 192.31.63.8
- 6. If this address is not in ARP cache, it broadcasts an ARP request on dept-2 Ethernet and finds the address as  $E_6$ .
- 7. An Ethernet frame addressed to  $E_6$  is built, packet is put in payload field, sent over Ethernet
- 8. When the Ethernet frame arrives at host-4, the packet is extracted from the frame and passed to IP software for processing.

**RARP:** RARP is used when an Ethernet address is given and corresponding IP address is to be found. This protocol a newly-booted work station to broadcast its Ethernet address and say: “**My 48-bit Ethernet address is 14.04.05.18.01.25. Does anyone know my IP address ?**”. The RARP server sees this request, looks up the Ethernet address in its configuration files and sends back the corresponding IP address.

**BOOTP:** RARP uses DA of all 1s to reach RARP server but such broadcasts are not forwarded by routers and therefore each n/w needs as RARP server. To get around this problem, BOOTP has been invented (defined in RFCs 951, 1048 and 1084). BOOTP uses UDP messages, which are forwarded over routers. It also provides a diskless work station with additional information, including the IP address of the file server holding the memory image, the IP address of the default router, and the subnet mask to use.

**The Interior Gateway Routing protocol: OSPF (Open Shortest Path First)**

The Internet is made up of a large number of Autonomous systems, in which each as is operated by a different organization and can use its own routing algorithm inside. A Routing algorithm with in an Autonomous system is called an **Interior Gateway Protocol**.

**Working:**

The algorithm is published in OPEN literature, and hence “O” in ‘OSPF’.

1. This protocol supports a variety of distance metrics, including physical distance, delay....etc..
2. It is a dynamic algorithm, which adapts to changes in the topology automatically and quickly.
3. It supports routing based on type of service.
4. It does load balancing by splitting the load over multiple lines.
5. It provides support for hierarchical systems.
6. It provides medium of security.
7. It provides a way to deal with routers that were connected to Internet via a tunnel

**OSPF supports 3 kinds of connections and networks:**

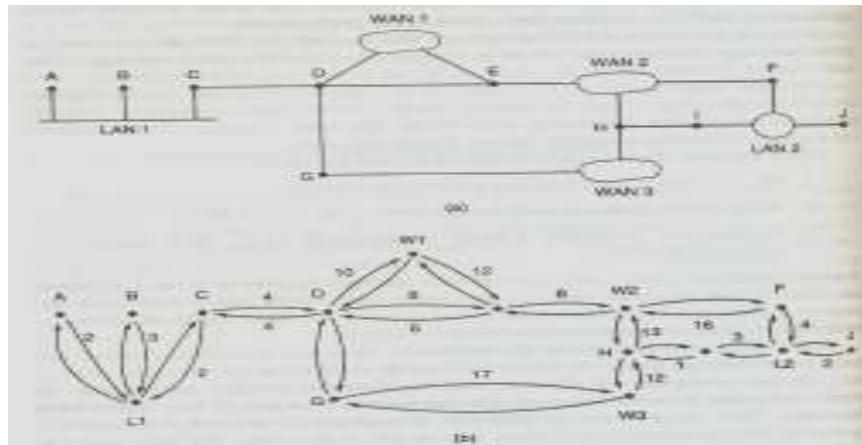
- 1) Point-to-Point lines between exactly 2 routers
- 2) Multi-access networks with broadcasting (**Eg:** most LANS)
- 3) Multi-access networks with broadcasting (**Eg:** most Packet Switched WANS)

**Multi-access network:** It is a n/w that can have multiple routers on it, each of which can directly communicate with all the others

**Working:-**

OSPF works by abstracting the collection of actual networks, routers and lines into a directed graph in which each arc is assigned a cost (distance, delay etc). It then computes the shortest path based on the weights on the arcs. A serial connection between two routers is represented by a pair of arcs, one in each direction. Their weights may be different. A multi-access network is represented by a node for the n/w itself plus a node for each router. The arcs from n/w node to the routers have weight ‘0’ and are omitted from the graph. What OSPF fundamentally does is represent the actual n/w like a graph and to compute the shortest path from every router to every other router.

An Autonomous System (AS) in the internet is large and so is divided up into numbered areas, where an area is an n/w or a set of contiguous n/w & and is a generalization of a subnet. Every 'AS' has a backbone area, called area-'0'. All areas are connected to the backbone, possibly by Tunnels (represented as an arc in the graph). The topology of areas and backbone are not visible outside the backbone.



Within an area, each router has the same link state database and runs the same shortest path algorithm. Its main job is to calculate the shortest path from itself to every other router in the area, including the router that is connected to the backbone, of which there must be at least one. The OSPF handles the service routing type by having 3 multiple graphs labeled using different metrics accordingly. So that separate routes are allowed for optimizing delay, throughput and reliability.

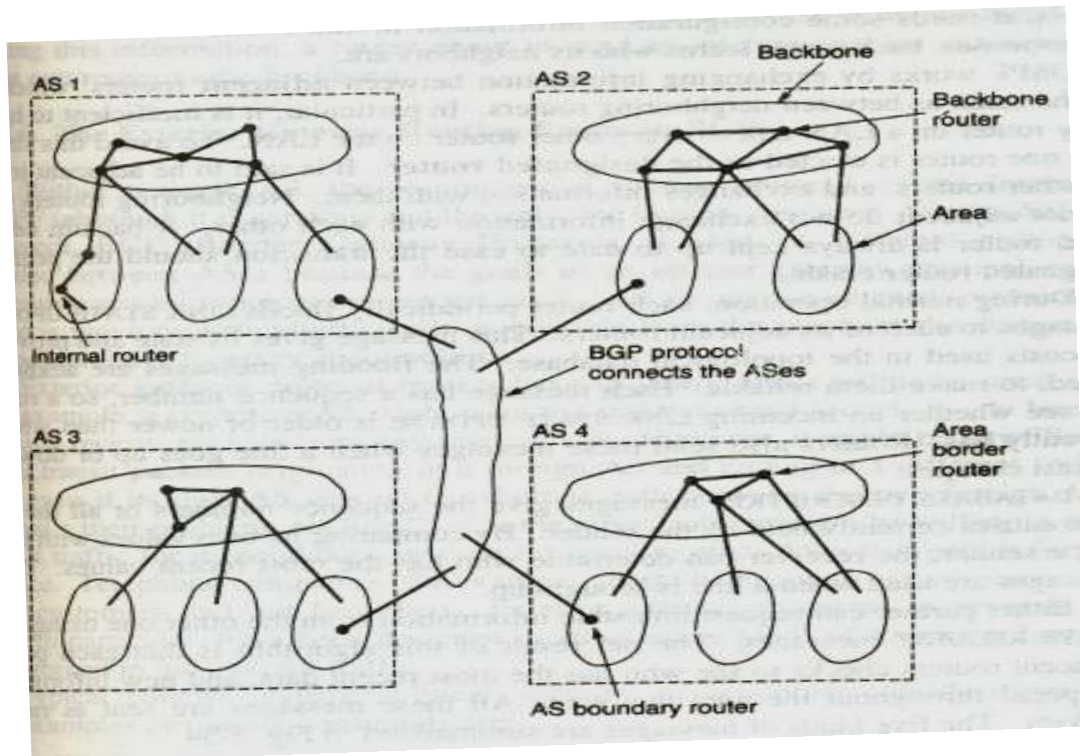
Graph Labeled with	Metric
COST	DELAY
COST	THROUGHPUT
COST	RELIABILITY

During Normal Operation, 3 kinds of routes may be needed:-

- **Intra-area:** These are easiest since the source router already knows the shortest path to the destination router.
- **Inter-area:** Always proceeds in 3 steps:-

- (a) Go from source to the backbone.
- (b) Go across the backbone to destination area.
- (c) Go to the destination

- **Inter-AS:** Packets are routed from source to destination being encapsulated or tunneled when going to an area whose only destination to the backbone is a tunnel.



#### **OSPF distinguishes 4 classes of routers:**

- 1) **Internal Routers:** These are wholly within one area.
- 2) **Area Border Routers:** These connect 2 or more areas
- 3) **Backbone Routers:** These are on the backbone
- 4) **AS Boundary Routers:** These talk to routers in other autonomous systems.

When a router boots, it sends **HELLO** messages on all of its point-to-point lines and multicasts then on LAN's to the group consisting of all other routers. OSPF works by exchanging information between **adjacent routers**. One router is elected as the **Designated Router** and it is said to be adjacent to all the other routers and exchanges information with them.

During normal operation, each router periodically floods **LINK STATE UPDATE** messages along with sequence numbers to each of its adjacent routers by which it can differentiate new from the old ones. This message gives its state and provides the costs used in topological database. Used when a line goes up/down. Each partner can request link state information from the other one using **LINK STATE REQUEST** messages, by which each pair of adjacent routers check to see who has the most recent data. All these messages are sent as raw IP packets. **DATA BASE DESCRIPTION** messages give the sequence numbers of all the link state entries currently held by the sender. Used when a line is brought up.

Message Type	Description
1) HELLO	Used to discover who the neighbors are
2) LINK STATE UPDATE	Provides the sender's costs to its neighbors.
3) LINK STATE ACK	Acknowledges link state update.
4) DATA BASE DESCRIPTION	Announces which updates the sender has.
5) LINK STATE REQUEST	Requests information from the partner.

Using Flooding, each router informs all the other routers in its area of its neighbors and costs. This information allows each router to construct the graph for its areas and compute the shortest path. In addition, the backbone routers accept information from area border routers in order to compute the best route from each backbone router other router. This information is propagated back to the area border routers, which advertise it within these areas. Using this information, a router about to send an inter area packet, can select the best exit router to the backbone.

### **The Exterior Gateway Routing Protocol: (BGP-Border Gateway protocol)**

An algorithm for routing between ASes is called an **Exterior Gateway Routing Protocol**. It moves packets as efficiently as possible from source to destination, using different policies, examples are:-

- 1) No transit traffic through certain ASes.
- 2) Never put Iraq on a route starting at pentagon.
- 3) Traffic starting or ending at IBM should not transit MICROSOFT.
- 4) Policies are manually configured into each BGP router 2 BGP routers are said to be connected if they share a common n/w. Networks are grouped into 3 categories depending on BGP traffic transit they are:-

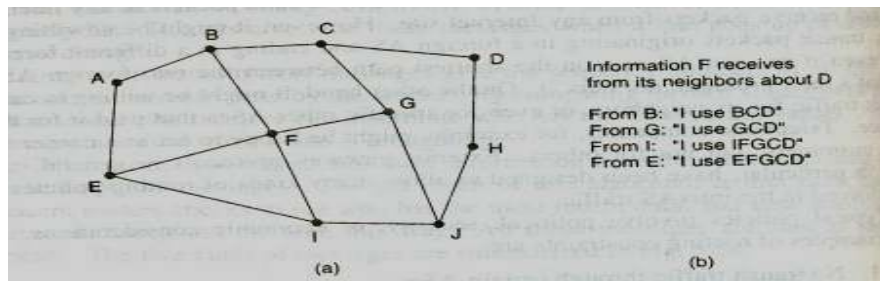
**Stub Networks:** Which have only 1 connection to the BGP graph.

**Multi-connected Networks:** These could be used for transit traffic, except that they refuse.

**Transit Networks:** Like backbones, willing to handle third party packets, possibly with some restrictions.

### **Working:**

Pairs of BGP routers communicate with each other by establishing TCP connections so that reliable communication is provided and all n/w details are hidden. BGP is fundamentally a distance vector protocol, which keeps track of the exact path used besides maintaining cost to each destination and its related to each of its neighbors.



From the above figure, consider F's routing table. Suppose that it uses the path FGCD to get to D. When the neighbors give it routing information, they provide their complete paths as from which the best path is chosen by 'F'. After seeing the neighbors information, 'F' discards the paths from I and E as they pass through 'F' itself. Any one of the paths from B or G is selected.

Neighbor	Route Used
B	BCD
E	EFGCD
G	GCD
I	IFGCD

### **The selection is done in this way:**

- Every BGP router contains a module that examines routes to a given destination and scores them, returning a number for the "distance" to that destination for each route.
- Any route violating a policy constraint automatically gets a score of " $\infty$  (infinity)".
- The router then adopts the route with the shortest distance.



**BGP easily solves even the count-to-infinity problem in this way:**

Suppose, 'G' crashes or the line 'FG' goes down 'F' receives routes from 3 remaining neighbors, which are BCD, IFGCD, EFGCD respectively from B, I, E from which FBCD chosen as its new route ( IFGCD, EFGCD pass through 'F' itself and so neglected.)

**# INTERNET MULTICASTING:**

IP supports Multicasting using class-D addresses. Each class-D address identifies a group of hosts. 28 bits are available for identifying groups, so over 250 million groups can exist at the same time. When a process sends a packet to a class-D address, a best efforts of the group addressed, but no guarantees are given. Some members may not receive.

Two kinds of group addresses are supported:

- 1. Permanent Addresses**
- 2. Temporary Addresses**

- 1. Permanent Addresses:** They are always there and don't have to be set up.

Some examples:

224.0.0.1	--	All systems on a LAN.
222.0.0.2	--	All OSPF Routers on a LAN.
224.0.0.5	--	All OSPF Routers on a LAN.
224.0.0.6	--	All designed OSPF routers on a LAN.

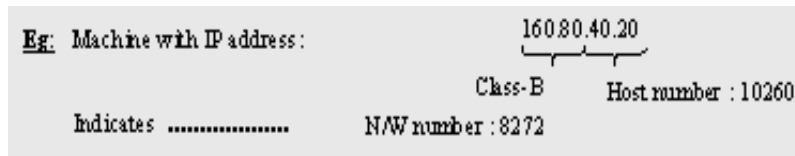
- 2. Temporary Addresses:** They must be created before they can be used. A process can ask its host to join a specific group or to leave the group. When the last process on a host leaves a group, that group is no longer present on the host.

About once a minute, each multicast router sends a hardware (**Eg: DLL**) multicast to the hosts on its LAN (284.0.0.1 address) asking them to report back on the groups their processes currently belong to. Each host sends back responses for all the class-D addresses it is interested in. These query and response packets use a protocol called **IGMP (Internet Group Management Protocol)**.

Multicast Routing is done using **SPANNING TREES**. Each multicast router exchanges information with its neighbors using a modified distance vector protocol in order for each one to construct a spanning tree per group covering all group members. Various optimizations are used to prune tree to eliminate routers and networks not interested in particular groups. The protocol makes heavy use of tunneling to avoid bothering nodes not in a spanning tree.

### # **MOBILE IP :**

Every IP address contains 3 fields, namely the class, the network number and the host number.



Routers all over the world have routing tables telling which line to use to get to network 160.80. Whenever a packet comes in with a destination IP address of the form 160.80.xxx.yyy, it goes out on that line.

**Problem:** If the machine with that address is carted off to some distant site [the packets for it will continue to be routed to its home LAN (or router)]. But the owner will no longer get E-mail and so on.

**Sol-1:** To give the machine a new IP address corresponding to its new location.

**Drawback:** Large no. Of people, programs, and databases would have to be informed of change

**Sol-2:** To have routers use complete IP addresses for routing instead of just class and network.

**Drawback:** Requires each router to have millions of table entries, at astronomical cost to Internet

### **To overcome these problems, some goals are desired which are to be met:**

1. Each Mobile host must be able to use its home IP address anywhere.
2. Software changes to the fixed hosts were not permitted.
3. Changes to the router software and tables were not permitted.
4. Most packets for mobile hosts should not make detours on the way.
5. No overhead should be incurred when a mobile host is at home.

**These goals are met with the following techniques:**

- Every site that wants to allow its users to roam has to create a home agent.
- Every site that wants to allow visitors has to create foreign agent.
- When a mobile host shows up a foreign site, it contacts the foreign host there and registers.
- The foreign host then contacts the user's home agent and gives it a care-of-address; normally the foreign agents own IP address.

**Working:**

- ❖ When a packet arrives at the user's home, it comes in at some router attached to the LAN.
- ❖ The router then tries to locate the host in the usual way, by broadcasting an ARP packet asking, for example, **"What is the Ethernet address of 160.80.40.20?"**
- ❖ The home agent responds to this query by giving its own Ethernet address.
- ❖ The router then sends packets for 160.80.40.20 to the home agent. It in turn, tunnels them to the care-of-address by encapsulating them in the payload field of an IP packet addressed to the foreign agent.
- ❖ The foreign agent de correlates and delivers them to the data link address of the mobile host.
- ❖ In addition, the home agent fives the Care- of – address to the sender, so future packets can be tunneled directly to the foraging agent.

**Gratuitous ARP :**

This is a trick used by the router to replace the Ethernet address of a mobile host with the Ethernet address of Home agent, cover the mobile host moves.

**ADVERTISEMENTS:**

The mobile host broadcasts a packet announcing its arrival and hoe that the local foreign agent responds to it and these broadcast messages are called **Advertisements**

**To locate Agents .....**

- Each agent is made to periodically broad cast its address and the type of services it is willing to provide (**Eg:** Home, foreign or both). When a mobile host arrives somewhere, it can just listen for these broad cast, called **ADVERTISEMENTS**.

**CIDR – Classless Inter Domain Routing:****Three Bears Problem:**

For most organizations, organizing the address space by a class-A network with 16million addresses is too big and by a class-c network with 256 addresses is too small but by a Class-B network with 16,536 is just right. This situation is called “**Three bears Problem**”.

**Problems in the Internet:**

- In reality, a class – B address is far too large for most Organizations
- As routers have to know about the N/Ws, Every router to maintain a table with all the N/W entries, one per N/W which requires more Physical storage and which is expensive.
- Various routing algorithms require each route to transmit its table’s periodically. The Larger the tables, the larger the routing instabilities.
- Using a deeper hierarchy for routing tables (i.e., having each IP address: country -> state-> cites -> N/W -> Host) requires more than 32 bits for IP addresses.
- All these problems can be stoned using CIDR, The basic idea behind this is to allocate the remaining Class-C network in Variable – sized blocks (almost 2 million N/Ws).

**Eg:** If a site needs 2000 addresses, it is given a block of 2048 addresses (8 contiguous class-C networks ) or If a site needs 8000 addresses, it is given a block of 8192 addresses (32 contiguous class- C networks)

In addition to using blocks of contiguous class-C networks as Units, the allocation rules for class ‘C’ addresses are implemented as :-

- The World was partitioned into 4 zones.
- Each zone is given a position of class – C address space

Address Range	Zone
194.0.0.0 to 195.255.255.255	Europe
198.0.0.0 to 199.255.255.255	North America
200.0.0.0 to 201.255.255.255	Central & South America
202.0.0.0 to 203.255.255.255	Asia & The pacific.

In this way, each region was given about 32 million addresses to allocate, with another 320 million class - C addresses from 204.0.0.0 through 223.255.255.255 reserved for future use. To overcome the routing table explosion more precisely, 114 CIDR uses a 32-bit mask attached to each routing table entry. When a packet comes in, its destination address is first extracted there the routing table is scanned entry by entry, masking the destination address and comparing it to the table entry 100 king for a match.

University	Needs addresses	Assigned addresses	Mask assigned	Binary addresses	Binary Mask
U-1	2048	194.24.0.0 to 194.24.7.255	255.255.248.0	1100 0010 0001 1000 0000 0000 0000 0000	1111 1111 1111 1111 1000 0000 0000
U-2	4096	194.24.16.0 to 194.24.31.255	255.255.240.0	1100 0010 0001 1000 0001 0000 0000 0000	1111 1111 1111 1111 0000 0000 0000
U-3	1024	194.24.8.0 to 194.24.11.255	255.255.252.0	1100 0010 0001 1000 0000 1000 0000 0000	1111 1111 1111 1111 1100 0000 0000

If a packet comes in addressed to 194.24.17.4 i.e., **1100 0010 0001 1000 0001 0001 0000 0100.....**

1. It is Boolean AND ed with U-1 mask to get **1100 0010 0001 1000 0001 0000 000 0000 0000** which doesn't match the U-1 base address.
2. The original address is ANDed with U-2 Mask to get **1100 0010 0001 1000 0001 0000 0000 0000** which matches the U-2 base address and so the packet is delivered.

### IP-V<sub>6</sub>:

The driving motivation for the adoption of a new version of IP was the limitation imposed by 32-bit address field in IP-V<sub>4</sub>. Reasons for inadequacy of 32-bit address include the following:

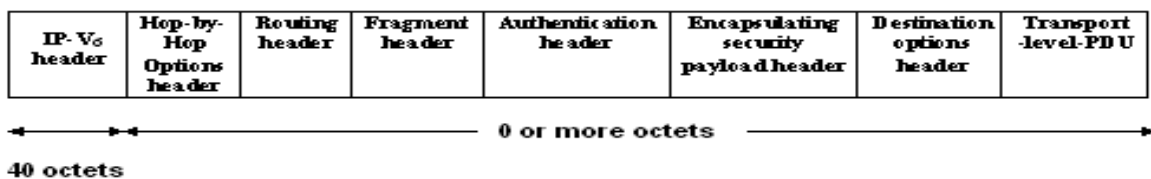
- ➔ 2-level structure of the IP address( **n/w-number, host number** )is wasteful of address space
- ➔ The IP addressing model generally requires that a unique n/w number be assigned to each assigned to each network whether or not it is actually connected to the internet.
- ➔ Networks are proliferating rapidly.
- ➔ Growth of TCP/IP usage into new areas will result in a rapid growth in the demand for unique IP addresses.
- ➔ It assigns a single IP address to each host instead of multiple.

IP-V <sub>6</sub> header	Hop-by-Hop Options header	Routing header	Fragment header	Authentication header	Encapsulating security payload header	Destination options header	Transport-level-PDU
--------------------------	---------------------------	----------------	-----------------	-----------------------	---------------------------------------	----------------------------	---------------------

←----- 0 or more octets -----→  
40 octets

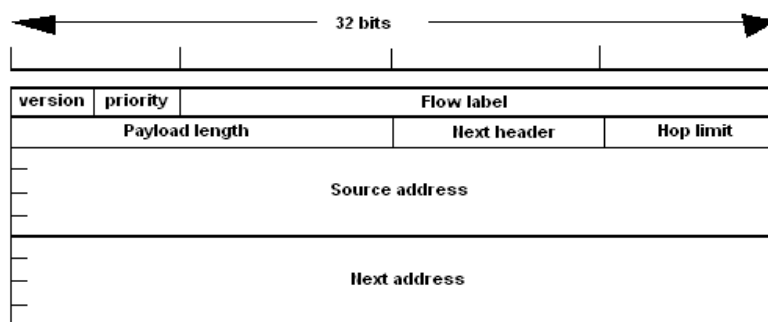
**Comparison of IPV<sub>6</sub> and IPV<sub>4</sub>:**

Parameter	IPV <sub>6</sub>	IPV <sub>4</sub>
1.Address	128-bit	32-bit
2.Data transfer elements	packets	datagrams
3.Optional header	separate	not separate
4.Address assignment	dynamic	static
5.data transfer speed	faster	slower
6.security	provides great security	provides smaller security
7.Addressing flexibility	more	less
8.support for resource allocation	yes <u>Eg:</u> real time video	no

**Extension Headers:**

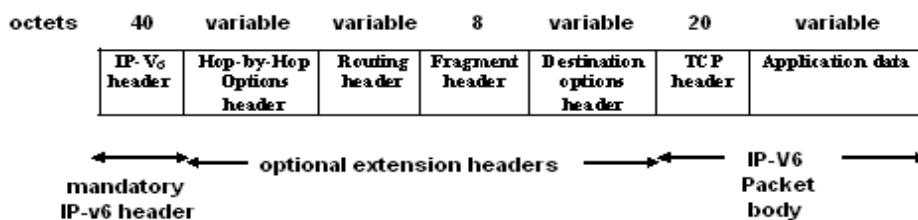
Extension Header	Description
Hop-by-Hop Options	Miscellaneous information for routers
Routing	Full or partial route to follow
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender's identity
Encrypted security Payload	Information about the encrypted contents
Destination options	Additional information for the destination

**IP-V<sub>6</sub> Header:** It has a fixed length of 40 octets and consists of following fields:



- ➔ **Version (4 bits):** internet protocol version number=6
- ➔ **Priority (4 bits):** is used to distinguish between packets whose sources can be flow-controlled and those that cannot. (0...7 = for transmission that are capable of slowing down in the event of congestion while 8...15 = real time traffic whose sending rate is constant)
- ➔ **Flow label ( 20 bits ):** may be used by a host to label those packets for which it is requesting special handling by routers with in a network.
- ➔ **Payload length(16-bits):** length of remainder of IPV6 packet following the header in octets.
- ➔ **Next header :** it tells which transport protocol handler (Eg: TCP,UDP) to pass the packet to.
- ➔ **Hop limit :** it is used to keep packets from living for ever.
- ➔ **Source address:** tells the address of the source from which the data is coming
- ➔ **Destination address:** tells the address of the destination to where it should go.

#### Extension headers:



**Hop-by –hop options header:** It is used for information that all routers along the path must examine. Datagrams using this header extension are called **JUMBOGRAMS**. So far, only one option has been defined i.e., support of datagrams exceeding 64k. The format is represented as follows:

Next header	0	194	0
Jumbo payload length			

**Routing Header:** It lists one or more routers that must be visited on the way to the destination. Both strict routing and loose routing are available, but they are combined. The format can be represented as:

Next header	0	Number of addresses	Next address
Bit – map			
1 to 24 addresses			

**Fragmentation header:** it deals with fragmentation. The header holds the datagram identifier, fragment number, and a bit telling whether more fragments will follow.

**Authentication header:** it provides a mechanism by which the receiver of a packet can be sure of who sent it.

**Encrypted security payload header:** it is used for packets that must be sent secretly.

**Destination options header:** it is intended for fields that need only be interpreted at the destination host.

### **The Network Layer in ATM Networks**

The ATM layer deals with moving cells from source to destination and involves routing algorithms and protocols within the ATM switches. The ATM layer is Connection-Oriented, both in terms of the service it offers and the way it operates internally.

When an application program produces a message to be sent, the message works its way down the protocol stack, having headers and trailers added and undergoing segmentation into cells. Eventually the cells reach the TC sub layer for transmission.

#### **Cell Transmission:**

The first step is Header Check Summing. Each cell contains a 5-byte header consisting of 4 bytes of virtual circuit and control information followed by a 1-byte checksum. The 8-bit checksum field is called **HEC (Header Error Control)**. This scheme corrects all single-bit errors and detects many multipoint errors as well. Once the HEC has been generated and inserted into the cell header, the cell is ready for transmission.

Transmission media comes in 2 categories:

- (a) **Asynchronous:** With this, a cell can be sent whenever it is ready for transmission. No time restrictions exist.
- (b) **Synchronous:** With this, cells must be transmitted according to predefined timing pattern of no data cell is available when needed, the TC sub layer must invent one. These are called idle cells.

The important task of TC sub layer is to match the ATM output rate to underlying transmission system rate and generating framing information for underlying transmission



**OAM CELLS:**

The non-data cells, which are used by ATM switches for exchanging control and other information necessary by keeping the system running. These are distinguished from data cells by having the first 3 header bytes by all zeros, something not allowed for data cells, and the 4<sup>th</sup> byte describes the nature of OAM cell. On the receiver's side, the idle cells are processed in the TC sub layer while the OAM cells are given to ATM layer.

**Cell Reception:**

On Input, the TC sub layer takes the incoming bit stream, locates the cell boundaries, verifies the headers, processes the OAM cells and passes the data cells up to the ATM layer.

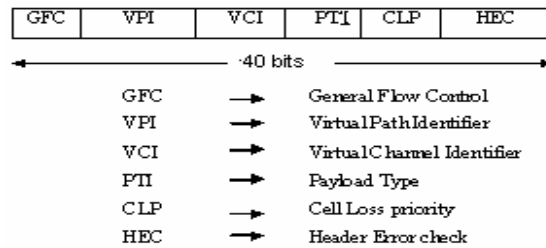
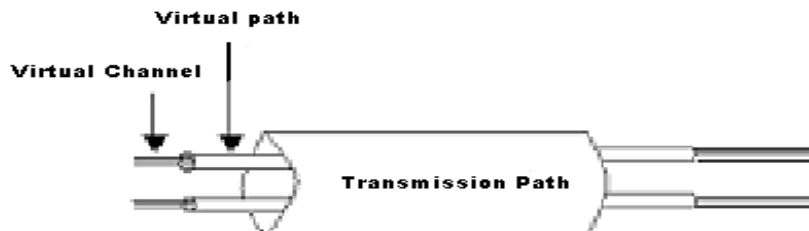
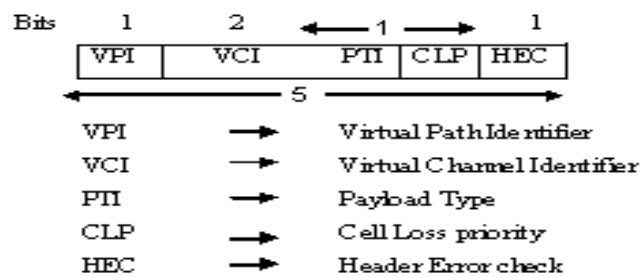
The hardest part is locating cell boundaries in the incoming bit stream. A cell is a sequence of  $53 \times 8 = 424$  bits with no "0111 1110" flag bytes to mark the start and End of a cell, and so cell boundaries can't be recognized under these circumstances. This problem can be dealt to some extent with SONET, in which the underlying physical layer provides help. With SONET, cells can be aligned with SPE (Synchronous Payload Envelope) pointer pointed to start of full cell.

If the physical layer provides no assistance for framing, HEC is used. On input, the TC sub layer maintains 40-bit shift register with bits entering on the left and exiting on the right. The TC sub layer then inspects the 40-bits to see if it is potentially a valid cell header and if so, the right most 8-bits will be a valid HEC over the left-most 32 bits and one stored in shift register. If it is not a valid cell header, all the bits in the buffer are shifted right one bit, causing one bit to fall off the end, and a new input bit is inserted at the left end. This process is repeated until a valid HEC is located. At that point, the cell boundary is known because the shift register contains a valid header. The trouble with this heuristic is that the probability of finding a valid HEC is  $1/256$ .

**# Cell Formats:**

In ATM layer, 2 interfaces are distinguished:

- **The UNI (User - Network - Interface)** - Boundary between a host and an ATM network.
- **The NNI (Network - Network - Interface)** - Line between o ATM switches.

**ATM layer header at UNI:****ATM layer header at NNI:**

Different values of PTI field are: -

Payload type	Meaning
000	User data cell, no congestion, cell type 0
001	User data cell, no congestion, cell type 1
010	User data cell, congestion experienced, cell type 0
011	User data cell, congestion experienced, cell type 1
100	Maintenance information between adjacent switches
101	Maintenance information between source and destination switches
110	Resource Management cell (used for ABR congestion control)
111	Reserved for future function

**# Connection Setup:**

ATM supports both permanent virtual circuits and switched virtual circuits. Connection setup is handled by the Control plane using a highly complex ITU protocol called Q.2931. Several ways are provided for setting up a connection.

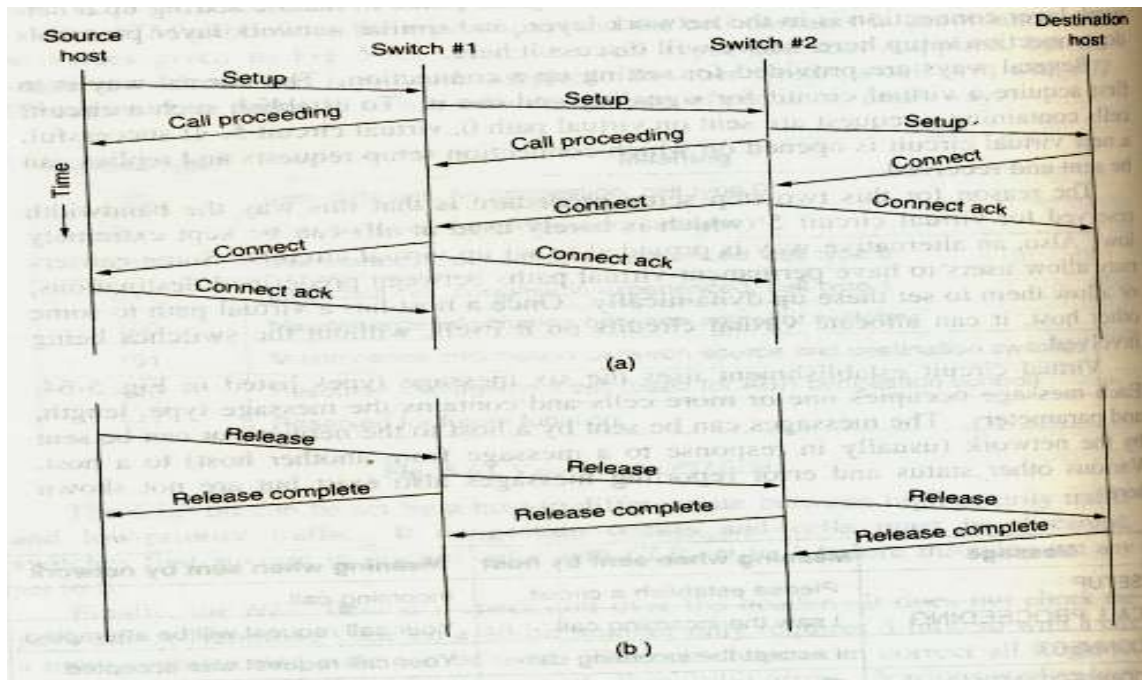
- **2-Step Setup Procedure**, in which a virtual circuit is acquired first, for signaling and then used.
- **Dynamical Setup**, in which permanent virtual paths are established between pre-defined destinations.

This virtual circuit establishment uses 6 message types, in which each message occupies one or more cells and contains message type, length and parameters. These can be transmitted between a host and a network.

Message	Meaning when sent by host	Meaning when sent by network
SETUP	Please establish a circuit	Incoming call
CALL PROCEEDING	I saw the incoming call	Your call request will be attempted
CONNECT	I accept the incoming call	Your call request was accepted
CONNECT ACK	Thanks for accepting	Thanks for making the call
RELEASE	Please terminate the call	The other side has had enough
RELEASE COMPLETE	Ack for RELEASE	Ack for RELEASE

**Connection Setup Procedure:**

1. A host sends a **SETUP** message including the destination address on a special virtual circuit, which travels through each hop to reach destination.
2. The network then responds with **CALL PROCEEDING** to acknowledge receipt of the request.
3. When **SETUP** finally reaches destination, it responds with **CONNECT** to accept the call.
4. The network then sends a **CONNECT ACK** message to indicate that it has received the **CONNECT** message.



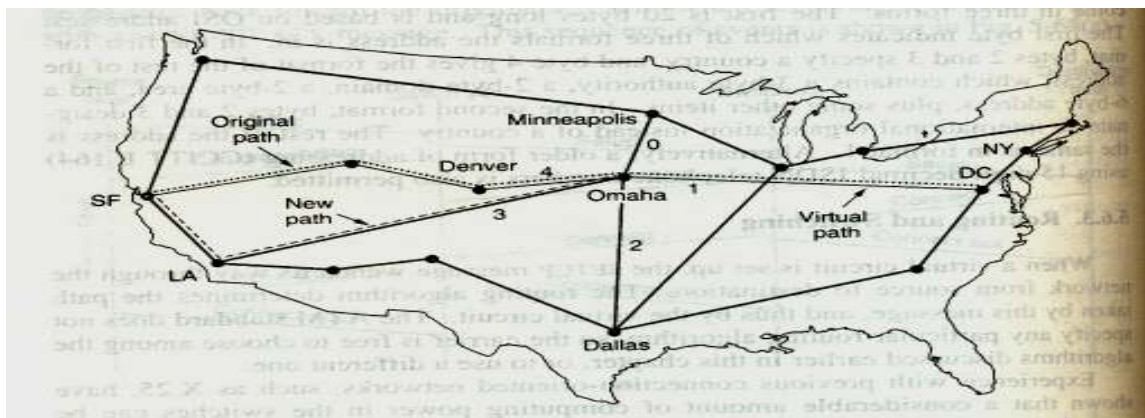
### Connection Release Procedure:

1. The host wishing to hang up just sends a **RELEASE** message.
2. When this message reaches to the other end, the circuit is released.

### # Routing and Switching:

When cells sent between a switch and a host, (or) in an interior switch ROUTING is done to route on the VPI field.

Routing of cells with in an Interior Switch ( Eg: OMAHA switch ):



1. For each of its 5 input lines, (0...4) it has a table, VPI-table, indexed by incoming VPI that tells which of the 5 outgoing lines to use and what VPI to put in outgoing cells.
2. For each outgoing line, the switch maintains a bitmap telling which VPIs are currently in use on that line.
3. When the switch is booted,
  - a. All entries in VPI table are masked as “NOT-IN-USE”
  - b. All bitmaps are marked to indicate that all VPIs are available.
4. Virtual circuits are full duplex in which each setup results in 2 entries
  - Forward traffic
  - Reverse traffic

Source	Incoming line	Incoming VPI	Destination	Outgoing line	Outgoing VPI	Path
NY	1	1	SF	4	1	New
NY	1	2	Denver	4	2	New
LA	3	1	Minneapolis	0	1	New
DC	1	3	LA	3	2	New
NY	1	1	SF	4	1	Old
SF	4	3	DC	1	4	New
DC	1	5	SF	4	4	New
NY	1	2	Denver	4	2	Old
SF	4	5	Minneapolis	0	2	New
NY	1	1	SF	4	1	Old

From the above table,

**NY → SF:** The cells should travel through line-1 to reach the destination-SF and is used for the first time. So, it is marked as (1,1) in forward traffic. The SH should use line-4 for reverse traffic and is used for first time. So, it is marked as (4,1) in Reverse traffic.

*Similarly, all the entries in the table*

### # Service Categories:

Class	Description	Example
CBR	Constant Bit Rate	T-1 circuit
RT-VBR	Variable Bit Rate ; real time	Real-time Video-Conferencing
NRT-VBR	Variable Bit Rate ; Non-real time	Multimedia email
ABR	Available Bit Rate	Browsing the web
UBR	Unspecified Bit Rate	Background file transfer

**CBR:** This class is intended to emulate a copper wire or optical fibre. Bits are put on one end and they come off the other end. No error checking, flow control, or other processing is done. All traffic is carried directly by an ATM system, with this class.

**VBR:** This class is divided into 2 subclasses, for real time and non-real time respectively.

***RT-VBR*** → This is intended for services that have variable bit rates continued with stringent real-time requirements.

***NRT-VBR*** → This is for traffic where timely delivery is important but a certain amount of jitter can be tolerated by the application.

**ABR:** This class is defined for bursty traffic whose bandwidth range is known roughly. This service avoids having to make a long term commitment to a fixed bandwidth. ABR is the only service category in which the network provides rate feedback to the sender, asking it to slow down when congestion occurs.

**UBR:** This class makes no promises and gives no feedback about congestion. UBR cells will be discarded without informing the sender, if congestion occurs. For applications that have no delivery constraints and want to do their own error control and flow control, UBR class is the best choice.

### **# Quality of Service:**

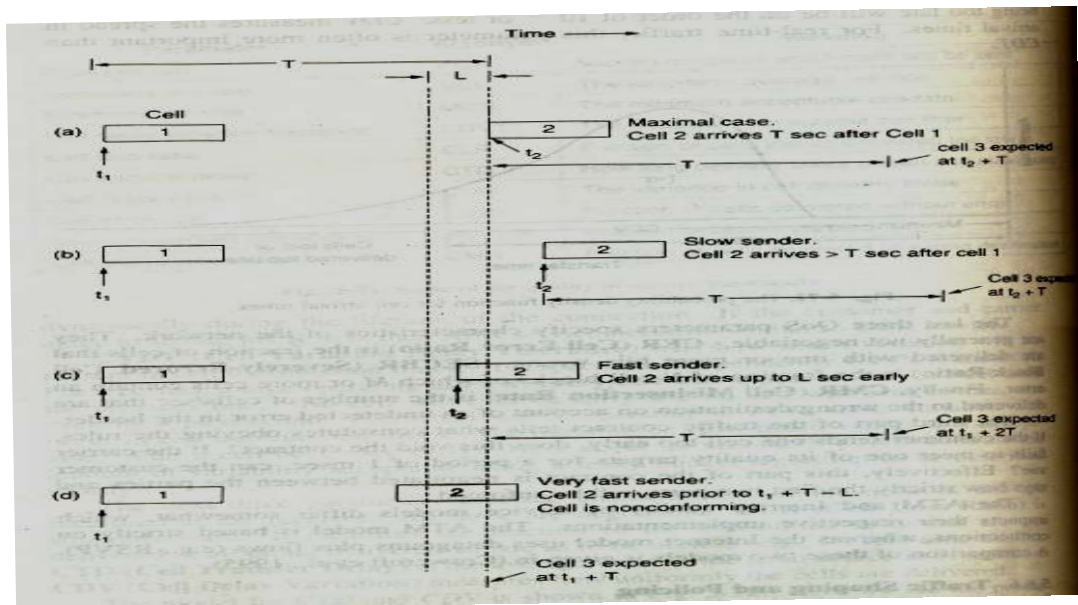
When a virtual circuit is established, both the transport layer and the ATM network layer must agree on a contract defining the service. ATM standard defines a number of QoS parameters to have concrete traffic between the customer and the carrier.

S.no	Parameter	acronym	Meaning
1	Peak Cell Rate	PCR	maximum rate at which cells will be sent
2	Sustained Cell Rate	SCR	the long term average cell rate
3	Minimum Cell Rate	MCR	the minimum acceptable cell rate
4	Cell Delay Variation Tolerance	CDVT	the maximum cell acceptable cell jitter
5	Cell Loss Ratio	CLR	fraction of cells lost or delivered too late
6	Cell Transfer Delay	CTD	how long delivery takes
7	Cell Delay Variation	CDV	the variance in cell delivery times
8	Cell Error Rate	CER	fraction of cells delivered
9	Cell Mis-insertion Rate	CMR	fraction of cells delivered to wrong destination



### # Traffic Shaping and Policing :

The mechanism for using and enforcing the QOS parameters is based on **Generic Cell Rate Algorithm (GCRA)**. It works by checking every cell to see if it conforms to the parameters for its virtual circuit. It has 2 parameters, which specify the maximum allowed arrival rate (PCR) and the amount of variation that is tolerable (CDVT). The reciprocal of PCR,  $T = 1/PCR$  is the minimum cell inter-arrival time.



**Fig-(a) →** Minimum Cell Inter arrival Time. ( Eg: PCR = 100,000 cells/sec  $\Rightarrow T = 10\mu\text{sec}$  )

**Fig-(b) →** A sender is always permitted to space consecutive cells more widely than  $T$ . any cell arriving more than  $T \mu\text{sec}$  after the previous one is confirming.

**Fig-(c) →** Senders tend to jump the gun. i.e., a cell arrives a little early (  $\leq t_1 + T - L$  ). It is Confirming but the next cell is still expected at  $t_1 + 2T$  (not  $t_2 + T$ ), to prevent the sender from transmitting every cell " $L\text{-}\mu\text{sec}$ " early, thus increasing PCR.

**Fig-(d) →** The cell arrives  $> L \mu\text{sec}$  early and so declared as Non-Confirming, and it is up to the carrier to treat this cell.

### Advantages of GCRA:

1. Used to make sure that Mean Cell Rate doesn't exceed SCR for any substantial period.
2. It provides a rule about which cells are confirming and which ones are not.

3. It shapes the traffic to remove some of the burstiness.

### **# Congestion control:**

ATM networks must deal with both long-term congestion, caused by more traffic coming in than the system can handle, and short-term congestion, caused by burstiness in the traffic.

Different strategies dealing with congestion fall into 3 categories.

1. Admission control
2. Resource reservation.
3. Rate-based congestion control.

#### **1. Admission control:**

When a host wants a new virtual circuit, it must describe the traffic to be offered and the service expected. The n/w can then check to see if it is possible to handle this connection without adversely affecting existing connections. Multiple potential routes may have to be examined to find one which can do the job. If no route can be located, the call is rejected.

#### **2. Resource Reservation:**

The resources are reserved in advance, usually at call setup time. The traffic descriptor contains peak cell rate [by which enough bandwidth can be reserved by n/w, along the path] and average cell rate [by which different circuits are multiplexed to meet the PCR].band width can be reserved by having setup message earmark bandwidths along each line it traverses.

#### **Rate - based congestion control:**

This was mainly developed to deal with congestion for ABR traffic. ABR congestion control is based on the idea that each sender has a current rate ACR(actual cell rate)that tells between the MCR and PCR.

- When congestion occurs, ACR is reduced ( not < MCR )
- When congestion is absent, ACR is increased ( not > PCR )

The 2 different solutions are

- ➔ Credit-based solution
- ➔ Rate-based solution.



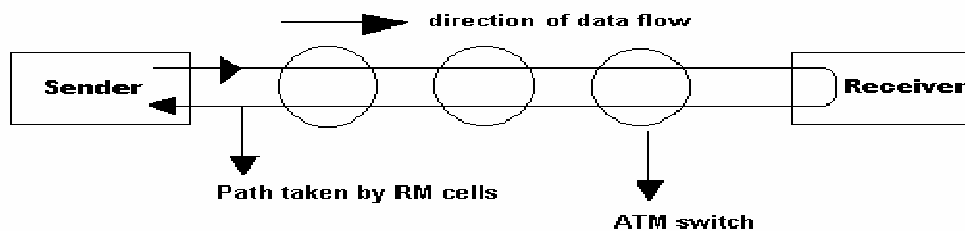
**Credit-based solution:** it was essentially a dynamic sliding window protocol that required each switch to maintain a credit per virtual circuit (Credit = no of buffers reserved for that circuit). As long as each transmitted cell had a buffer waiting for it, congestion could never arise.

**Drawbacks:**

- ➔ so many buffers are to be reserved in advance.
- ➔ The amount of overhead and waste required was thought to be too much.

**Rate-based solutions:**

The basic model is that after every K data cells, each sender transmits along the same path as the data cells, but is treated specially by the switches along the way. When it gets to the destination, it is examined, updated, and set back to the sender. The full path for RM cells is shown in below figure:



Each RM cell sent contains the rate at which the sender would ER (explicit rate). As the RM cell passes through various switches on the way to the receiver, those that are congested may reduce ER. No switch may increase it reduction can occur either in forward or in reverse direction when the sender gets the RM cell back, it can then see what the min. acceptable rate is acc to all the switches, along the path. It can then adjust ACR (if needed), to bring it into line with what the slowest switch can handle.

**ATM LANS:** The main task is to provide connection-less LAN service over a connection-oriented ATM. So, different techniques were proposed:

**Sol-1:** A connection-less server is introduced into the n/w with which a host initially set up a connection and all packets are sent to this server, for forwarding.

**Drawback:**

- ➔ doesn't use the full bandwidth of ATM n/w.

→ connection-less server can easily become a bottleneck.

**Sol-2:** This was proposed by ATM forum. In this, every host has a (potential) ATM virtual circuit to every other host. These virtual circuits can be established and released dynamically as needed, or they can be permanent. To send a frame, the source host first encapsulates the packet in the payload field of an ATM AAL message and sends it to the destination.

**Problem:**

Difficult to tell which IP belongs to which virtual circuit., this problem is solved by the introduction of new server called the LES(LAN emulation server), which returns the ATM address to the machine, requesting it.

**Sol-3(IETF method ):** Functionality is same as in Sol-2, but the LES server is called **ATMARP** server. In this method, a set of ATM hosts can be grouped together to form a logical IP subnet. Each LIS has its own ATMARP server, in which LIS acts like a virtual LAN. Hosts on the same LIS may exchange IP packets directly, but hosts on different ones required to go through a router.