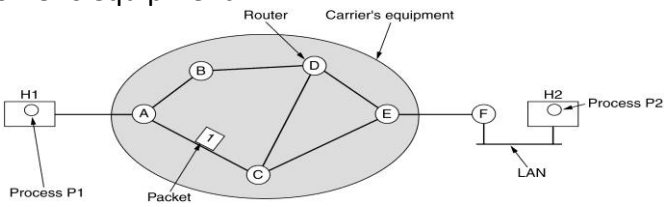
NETWORK LAYER

The network layer is concerned with distributing packets from the source to the destination. To reach to the destination, a packet might have to visit many hops at intermediate routes. To achieve its goals, the network layer must know about the topology of the communication subnet and choose appropriate paths through it. It must also take care to choose routes to avoid overloading some of the communication liens and routers.

Design Issues :

1.Store and Forward Packet Switching : The major components of the network are

1. carrier's equipment : routers connected by transmission lines.
2. customer's equipment

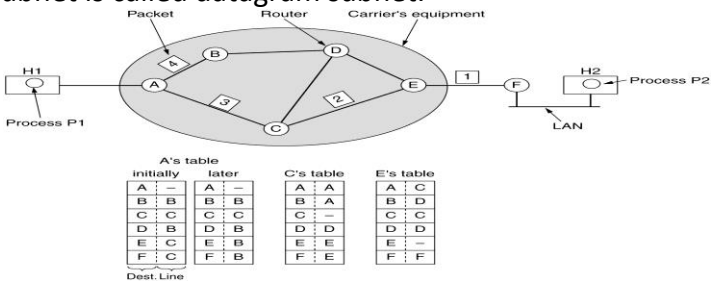


Host h1 is directly connected to one of the carrier's routers A, by leased line. H2 is on some other LAN with F. A host with a packet to send first it transmits to the nearest router, either on its own LAN or by using a point to point link to the carrier. The packet is stored there until it is fully arrived so the checksum can be verified. Then it is forwarded to the next route along the path until it reaches to the destination. This mechanism is store and forward device.

2. Services provided to the Transport Layer : The n/w layer provides services to the transport layer at the interface. It is the interface between the carrier and customer. The carrier often has control of the protocols and interfaces up to the n/w layer.

1. services should be independent of subnet topology
2. Transport layer should be shielded from the number, type and technology of the subnets present.
3. The n/w addresses must be available to the transport layer .

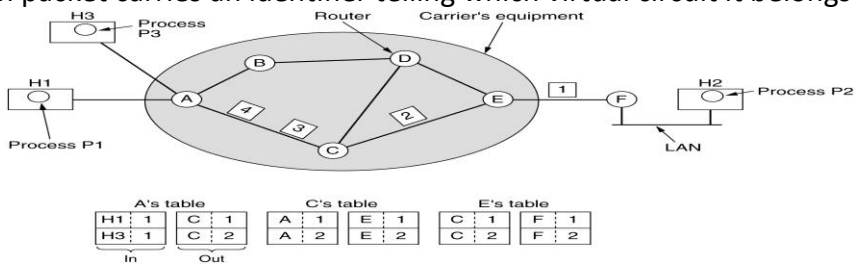
3. Implementation of Connection less Service : All packets are injected into the subnet and routed individually. No advance set up is required. The packets are called datagrams and the subnet is called datagram subnet.



The process P1 has a long message to process P2 on host 2. P1 hands over the message to the transport layer to deliver it to P2. The transport layer code runs on h1, within the operating system. It adds a transport header to the front of the message and hand over the result to the n/w layer. Suppose if the message is 4 times longer than the packet size, then the n/w layer has to break the packet into 4 packets. Every router has an internal routing table to specify where to send packets for each possible destination. Each entry will have a destination station and the outgoing line to be used.

Here A has only two outgoing lines B and C. Packets 1 is forwarded to E and then to F. Packet 2 and 3 follow the same route. Because of huge traffic in the old path, A might decide to send the packet to 4 via another router. This is a change in path.

4. Implementation of Connection-Oriented Service : When a connection is established, a route from the source machine to destination machine is chosen as part of the connection set up and routing tables are stored in side the routers. The same route is used for the entire traffic flowing over the connection. With connection oriented service, each packet carries an identifier telling which virtual circuit it belongs to.



Host H1 has established a connection 1 with host H2. Therefore it is sent to router C with connection identifier 1. similarly c routes packet to E. suppose if H3 also wants to establish a connection to H2. It chooses connection identifier 1 and tells the subnet to establish a virtual circuit. This gets a second entry in the table. A can easily distinguish packets from H1 and H3. but c can not identify. Therefore A assigns a different connection identifier to the outgoing traffic for the second connection.

5. Comparison of Virtual-Circuit and Datagram Subnets :

Issue	Datagram subnet	Virtual-circuit subnet
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

Routing Algorithms :

Establishing a suitable path, between source and destination by following effective routing measurements.

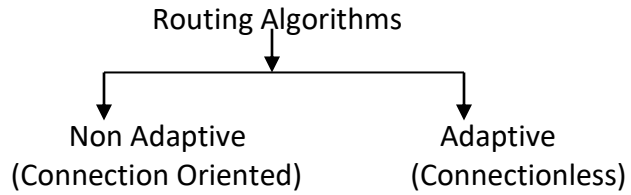
Routing Properties : To prepare a routing algorithm, it must follow the following principles.

1. It must be correct
2. It must be simple
3. It must be robust
4. It must have stability
5. It must be fair
6. It must be optimal

Routing measurements : Path can be established by measuring any of the case

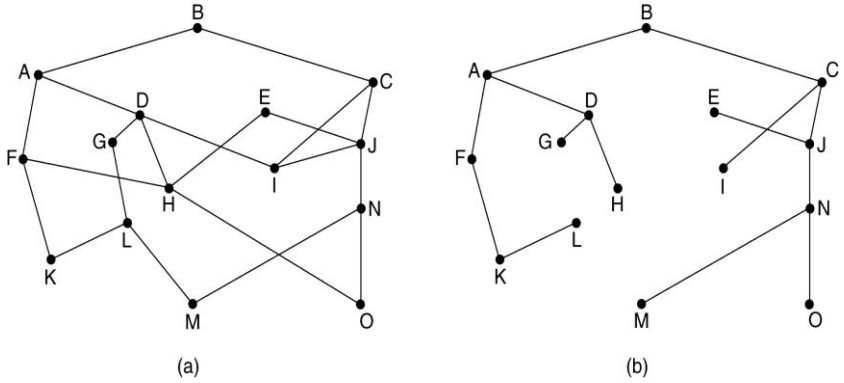
1. optimal distance in kms from source to destination
2. optimal time in msec to reach a packet from source to destination
3. optimal number of hops(intermediate routers) between source and destination.

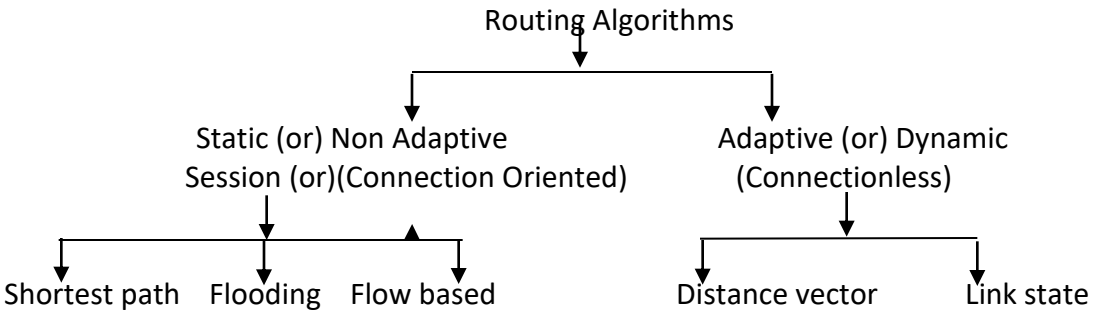
Routing algorithms are classified into two broad categories, Adaptive and Non Adaptive algorithms.



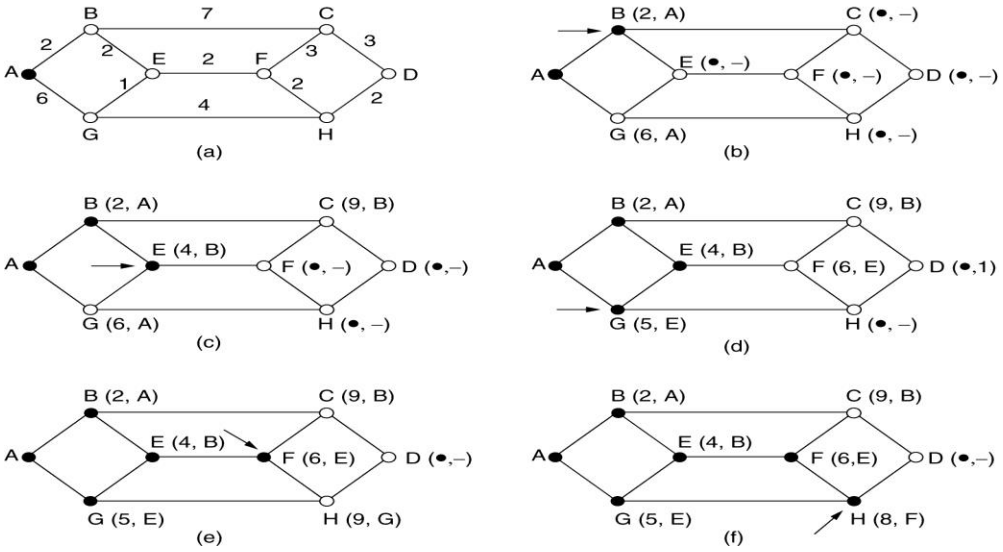
Non Adaptive algorithms do not base their routing decisions based on measurements or estimates of the current traffic and topology. All the routing is predefined during network booting. Some times this is also known as static routing. Adaptive routing algorithms change their routing decisions by following the current traffic and topology. Some times this is also known as dynamic routing.

Optimality Principle: It provides unique optimal path from source to destination. In a given subnet if we provide the set of all routes from all sources to a given destination to form a tree routed at the destination. Such type of tree is called sink tree or spanning tree. It does not contain any loops, so each packet will be delivered to within a finite and bounded number of hops.





Shortest Path Routing : This is also known as Dijkstras routing algorithm. The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line. To choose a route, this algorithm finds the shortest path between the pair of stations. This algorithm calculates distance between the routers.

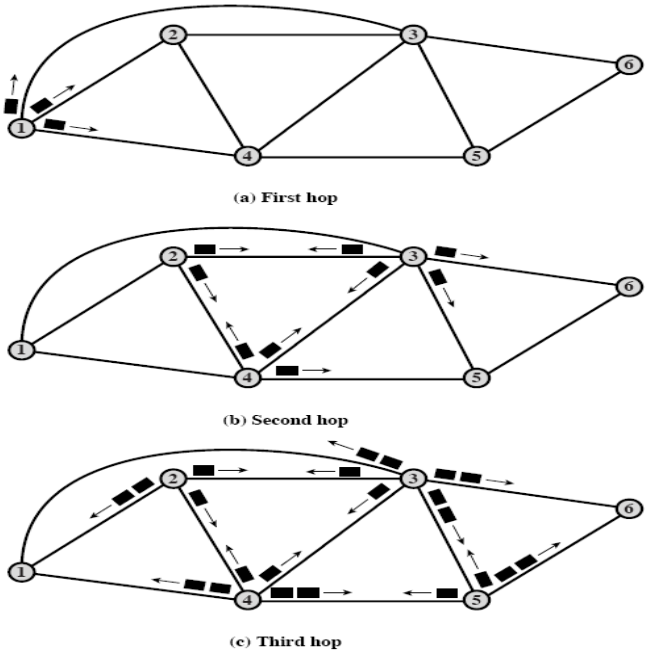


In this example, we want to find a path A to D. we start at node A as source, indicated by a filled circle. Examine each of the modes adjacent to A are labeling each one with distance to A. since B is the optimal one, examine all nodes adjacent to B. the nodes which have not been examined are labeled as (infinity,-). And all the examined nodes cost will be represented as (2,A) where the first co ordinate represent the distance covered till now and the previous station name. This means the cost from node A to node B is 2 and it came from node A. This process will continue till we reach to destination D.

Flooding: Every incoming packet is sent out one very out going line except the one it came from. It obviously generates vast number of packets. Here the routing measurement is hop count between source and destination. After visiting each router the value will be decremented, when it reach to zero it must be with the destination or

else the packet is going to be discarded by the intermediate routers itself. This algorithm is preferable when no routing tales are prepared at the initial stage.

The working method is explained in the following diagram. When we want to route a packet from A to D, First we have to create two duplicate packets of the original, since A has two alternative paths that can lead to D. so at the first stage there are two copies. In the next stage, again stations B and E create two more duplicate packets each, to distribute to their neighbors C and F. So at the second stage 4 more copies are created. All duplicate packets will be sent to destination. Whichever the packet first reaches to the destination, it will be accepted and all remaining duplicates are rejected by checking packet sequential numbers using checksum algorithm.



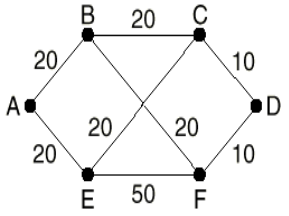
(a) First hop

(b) Second hop

(c) Third hop

Flow Based Routing: First two algorithms considered only topology of the subnet into account. They do not consider the load. Suppose if there are two alternative paths to reach from A to D, if there is a heavy traffic in the line AB always, then it may be better to choose path AE to route the packet.

The basic idea behind the analysis is that for a given line if the capacity and average flow are known, it is possible to compute the mean packet delay on that line from queuing theory formula. Let us take an example subnet which is drawn below. To use this technique certain information must be known in advance. First the subnet topology must be known. Second the traffic matrix, Fij must be given. Third the line capacity matrix C_{ij} must be given specifying the line capacity in bps. Finally a routing algorithm must be chosen.



(a)

Destination

Source	A	B	C	D	E	F
A		9 AB	4 ABC	1 ABFD	7 AE	4 AEF
B	9 BA		8 BC	3 BFD	2 BFE	4 BF
C	4 CBA	8 CB		3 CD	3 CE	2 CEF
D	1 DFBA	3 DFB	3 DC		3 DCE	4 DF
E	7 EA	2 EFB	3 EC	3 ECD		5 EF
F	4 FEA	4 FB	2 FEC	4 FD	5 FE	

(b)

Fig. 5-8. (a) A subnet with line capacities shown in kbps. (b) The traffic in packets/sec and the routing matrix.

i	Line	λ_i (pkts/sec)	C_i (kbps)	μC_i (pkts/sec)	T_i (msec)	Weight
1	AB	14	20	25	91	0.171
2	BC	12	20	25	77	0.146
3	CD	6	10	12.5	154	0.073
4	AE	11	20	25	71	0.134
5	EF	13	50	62.5	20	0.159
6	FD	8	10	12.5	222	0.098
7	BF	10	20	25	67	0.122
8	EC	8	20	25	59	0.098

Fig. 5-9. Analysis of the subnet of Fig. 5-8 using a mean packet size of 800 bits. The reverse traffic (*BA*, *CB*, etc.) is the same as the forward traffic.

Consider the full duplex subnet. The weights on the arcs give capacities. Secondly, the traffic matrix has an entry for each source and destination pair. It also shows the route to be chosen and traffic in between them. By using the above information we can calculate the arrival rate of each line. Let us assume that the mean packet size is $1/\mu$ =800 bits. With this information we can calculate mean time delay of each line that is

$$T = 1/(\mu C - \lambda)$$

Where μ is mean packet size

λ is arrival rate of packets in the subnet

After estimating time delays for each line, we calculate the weight of each line that is

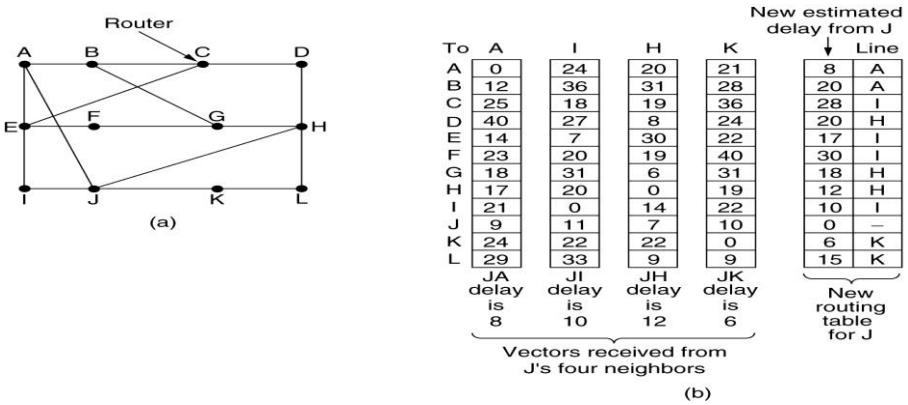
$$W = \lambda_i / \Sigma\lambda_i$$

Now the following table shows the flow based routing table. Always consider either forward direction or reverse direction while calculating arrival rates or prepare a routing table.

S.No	Line	λ_i(pkt/sec)	C(Kbps)	μC_i (pkts/sec)	T_i (msec)	Weight
1	AB	14	20	25	91	0.171
2	BC	12	20	25	77	0.146
3	CD	6	10	12.5	154	0.073
4	AE	11	20	25	71	0.134
5	EF	13	50	62.5	20	0.159
6	FD	8	10	12.5	222	0.098
7	BF	10	20	25	67	0.122
8	EC	8	20	25	59	0.098

Distance Vector Routing: This generally use dynamic routing algorithms. Distance vector routing algorithms operate by having each router maintain a table giving best known distance to each destination and which line to use to go to destination. These tables are updated by exchanging information with the neighbors. It is some times called Bellman-Ford and Ford Fulkerson Routing algorithm.

In distance vector routing, each router maintains a routing table indexed by, and containing one entry for each router in the subnet. This entry contains two parts: the preferred outgoing line to use to go the destination, and an estimate of the time or distance to that destination. Here the metric used is number of hops, time delay in msec or total number of packets queued along the path. Among these we prefer timed delay in msec. If the metric is delay, the router can measure it directly by a special ECHO packet. The information must be updated for each T_{msec}.



Consider the above subnet, now j is the source station. If we have to route packets from j to other destination, we can route them through neighbors only. The neighbors of J are A,I, H, K. So

1. Identify the neighbors.
2. Calculate time delays for source to neighbors
3. Then calculate time delays to destination from all neighbors.
4. Select the optimal one among the above.

Ex: JA=8msec JI=10 msec JH=12 msec JK=6msec

To route a packet from A to G

1. j→A→G = 8+18 =26
2. j→I→G = 10+31=41
3. J→H→G = 12+6=18
4. J→K→G = 6+31=37

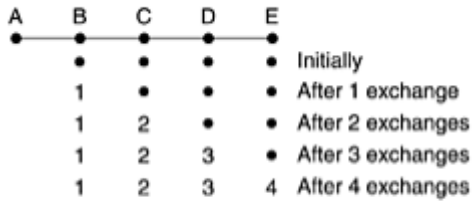
JHG is path is optimal. The algorithm has two problems, count to infinity and split horizon hack.

Count to infinity problem:

1. One of the important issue in Distance Vector Routing is County of Infinity Problem.
2. Counting to infinity is just another name for a routing loop.
3. In distance vector routing, routing loops usually occur when an interface goes down.
4. It can also occur when two routers send updates to each other at the same time.
5. Good news propagates very rapidly in this routing
6. Bad news propagates very leisurely in this routing

Example:

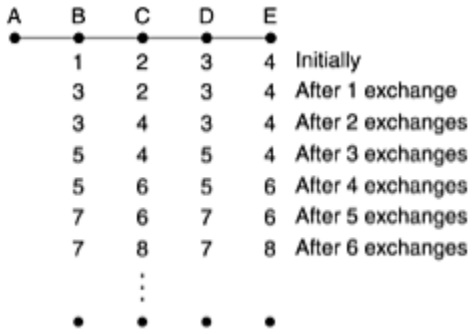
Example for Good news Propagation



In this routing, delay metrics is the number of hops.

1. Suppose router A is down initially and all other routers know this and they have recorded the delay to A is infinity.
2. When A comes up, in the first exchange b modifies its hop count as 1
3. In the second exchange C modifies its hop count to 2
4. In the third exchange D modifies its hop count to 3
5. In the fourth exchange, E modifies its hop count to 4

Example for Bad news Propagation



1. Since the router A is up, all other routers modify their hop count as 1, 2, 3, and 4.
2. Now imagine that the link between A and B is cut
3. At this time, B corrects its table.
4. After a specific amount of time, routers exchange their tables, and so B receives C's routing table.
5. Since C doesn't know what has happened to the link between A and B, it says that it has a link to A with the weight of 2 (1 for C to B, and 1 for B to A -- it doesn't know B has no link to A).
6. B receives this table and thinks there is a separate link between C and A, so it corrects its table and changes to 3 (1 for B to C, and 2 for C to A, as C said).
7. Once again, routers exchange their tables.
8. When C receives B's routing table, it sees that B has changed the weight of its link to A from 1 to 3, so C updates its table and changes the weight of the link to A to 4 (1 for C to B, and 3 for B to A, as B said).
9. Again in the 3rd exchange, both B and D increments their hop count to 5.

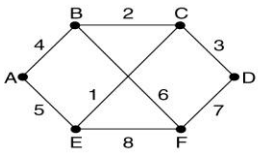
10. This process loops until all nodes find out that the weight of link to A is infinity.
11. This situation is shown in the above figure
12. In this way, Distance Vector Algorithms have a slow convergence rate.

One way to solve this problem is to maintain maximum number of routers as 15. Therefore when the hop count reached to 16 means actually there was not a path to A. Every router fooled by the other routers.

Link State Routing :

1. Discover its neighbors and learn their network addresses.
2. measure the delay or cost to each of its neighbors.
3. construct a packet by using the above information.
4. send this packet to all other neighbors.
5. compute the shortest path to every other router.

Once the information needed for the exchange has been collected, the next step is for each router is to build a packet containing all the data. The packet starts with the identity of the sender, followed by a sequential number and age(life time). All neighbor stations will be given entries in the routing table and their time delays are also involved.



(a)

Link State Packets													
A		B		C		D		E		F	
Seq.		Seq.		Seq.		Seq.		Seq.		Seq.	
Age		Age		Age		Age		Age		Age	
B	4	A	4	B	2	C	3	A	5	B	6
E	5	C	2	D	3	F	7	C	1	D	7
		F	6	E	1			F	8	E	8

(b)

Consider the above subnet by router B. Each row in the below table corresponds to recently arrived but nit yet fully processed link state packet. The table records from where the packet is originated, its seq no, age and the data. In addition to this, there ae send and ack flags for each B's neighbors A,C, and F. The send flags means that the packet 'N' just sent on the indicated line. The ack flags means it must acknowledge to those lines.

			Send flags			ACK flags			
Source	Seq.	Age	A	C	F	A	C	F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

The link state packet from A arrived directly, so it must be sent to C and F and acknowledged to A, as indicated by the flag bits. Similarly the packet from F has to be forwarded to A and C and Ack to F. similar case for entry C also.

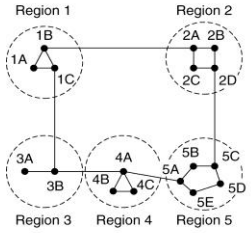
However the situation with the third packet, from E is different. It arrived twice, once via EAB and once EFB. So it has to sent only to C but acknowledged to both A and F. The same case for packet from D also.

Hierarchical Routing : As networks grow in size, the router routing tables grow potentially. Lot of router memory is occupied by ever increasing length of tables and

even more CPU time is required to scan them. So for the larger networks, routing must be hierarchical only like in telephone network.

When hierarchical routing is used, the routers are divided into regions, with each router knowing all the details about how to route a packet to destination with its own region, without knowing anything about internal structure of other regions. For huge networks two level hierarchy may be inefficient. It may be necessary to group the regions into clusters, clusters into zones and the zones into groups and so on. Here we prefer number of hops as routing measurement.

Let us consider the following subnet with two level hierarchy with five regions. The full routing table for router 1A has 17 entries. But when routing is done hierarchically, there are entries for all the local routers, but to all other regions, there will be a single entry. Therefore it saves time, space and bandwidth.



(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)

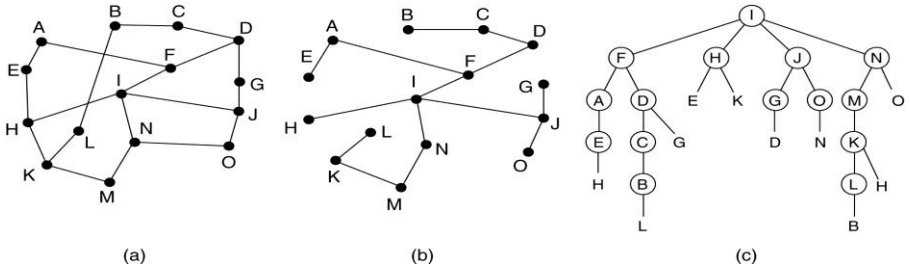
Hierarchical table for 1A

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

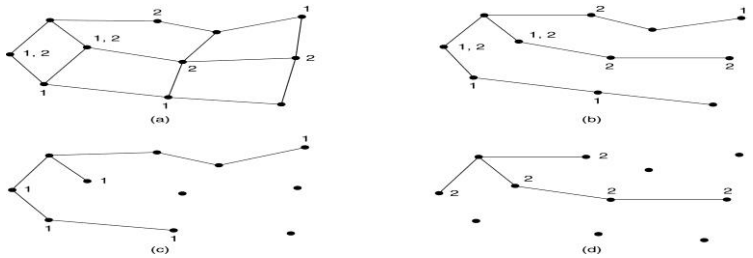
(c)

Broadcast Routing : For some applications, hosts need to send messages to many all other hosts. For example service distributing weather reports, stock market updates, or live radio programs. Sending a packet to all destinations simultaneously is called broadcasting. Broadcasting can be applied in several ways.

1. Sending a distinct packet to each destination. But it wastes lot of bandwidth and it also needs source to have a list of all destinations.
2. Multi destination Routing: Each packet contains a list of destinations or a bit indicating the desired destinations. When a packet arrives at a router, the router checks all the destinations to determine the set of output lines that will be needed. The router generates a new copy of the packet fore ach output line to be used and includes in each packet, only those destinations that are to use this line.
3. Sink tree or spanning tree : It is a subset of subnets that include all the routers but contains no loops. This methods makes the excellent use of bandwidth. By following this sink tree, a reverse path forwarding tree is established
4. Flooding : If none of these methods applicable, then flooding can be used. It consumes too much bandwidth by generating duplicates at each and every stage(hop).



Multicast Routing : For some applications, there is a need to send messages only to a group of applications. Sending a message to such a group is called multi casting. To do multicasting, group management is required. Some way is required to create and destroy groups, and for processes to join and leave groups. To do multicasting, each router computes a spanning tree converting all other routes in the subnet.



When a process sends a multicast packet to a group, the first router examines its spanning tree and prunes it, by removing all lines than do not lead to hosts that are members of the group. The above diagrams show for spanning trees fro groups 1 and 2.

Routing for Mobile hosts: Hosts that never move are said to be stationary. They are connected to the network by copper or fiber optics. The other two kinds of hosts are **migratory hosts** who are basically stationary hosts who move from one fixed site to another site from time to time. **Roaming hosts** actually maintain their connections as they move round. Therefore we use the term mobile host for the above two kinds.

All hosts are assumed to have a permanent home location that never changes. The routing goal of these hosts is to make it possible to send packets to mobile hosts using their home addresses and receive packets efficiently wherever they may be.

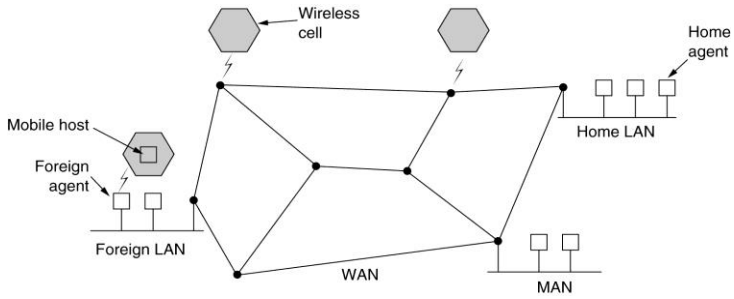
The entire world is divided into small regions. Let us call them areas, when an area is typically a wireless LAN or cell. Each area has one or more foreign agents, which is a visiting area of all mobile users who maintain their data there. Each area, has a home agent which keeps track of hosts whose home is in this area, but are currently visiting another area.

When a new host enters into an area, host computer must register itself with the foreign agent there. The registration procedure is

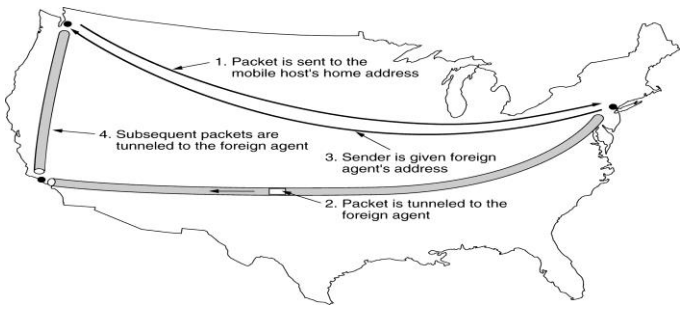
1. Periodically each foreign agent broadcasts a packet by announcing its existence and address. A newly arrived mobile host may wait for one of these messages. If

it did not receive any such kind of message, then the mobile itself broadcasts a packet by asking “Is there be any foreign agent?”

2. Mobile host registers with the foreign agent, by giving its home address, current data link layer address and some security information.
3. The foreign agent contacts the mobile host’s home agent to know about this user. This message will have the n/w address and security information of the mobile user of the foreign agent.
4. The home agent verifies the security information. If every thing is ok then it asks the foreign agent to proceed.
5. When the foreign agent gets the ack from the home agent, it makes an entry in its tables and informs the mobile host that it is now registered.
6. In the same way, when a host leaves from an area, same steps must be happened to do deregistration.



To send a packet from source to destination,



1. Packet is sent to the home agent on its home LAN
2. Then the Home agent encapsulates the packet in the payload of an outer packet and sends it to the foreign agent. (Tunneling)
3. After getting the encapsulated packet, the foreign agent removes the original packet from the pay load field and sends it to the mobile host as a data link frame.
4. The home agent tells the sender to send packets to the mobile host by encapsulating them in the payload of packets explicitly addressed to the foreign agent, instead of sending them to the mobile host’s home agent.

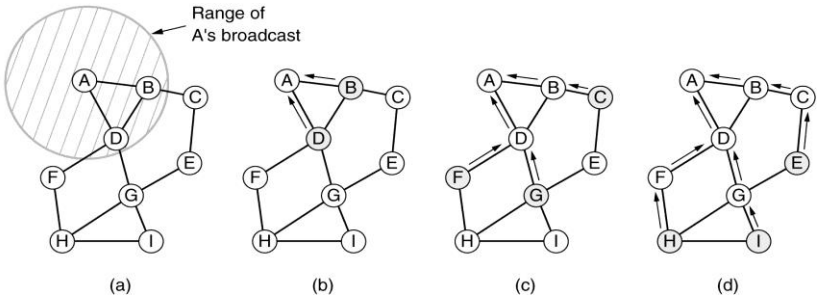
5. Therefore, subsequent packets can be routed directly to the mobile host via foreign agent, bypassing the home agent.

Routing in Mobile Ad Hoc Networks: In the above mechanism, only hosts are mobile but the routers are fixed. This is entirely different from the above. Here routers also mobile. The usage of MANETS is

1. Military vehicles on a battle field with no existing infrastructure.
2. A fleet of ships at sea
3. Emergency workers at an earthquake that destroyed the infrastructure.
4. A gathering of people with notebook computers in an area of lacking 802.11

Each node consists of a host and a router on the same computer. Networks of nodes that are near each other are called Mobile Ad Hoc Networks. In MANETS, the topology will change often and validity of paths can also change spontaneously. There are a variety of routing algorithms. Adhoc On Demand Distance Vector Routing is one among them. It is a far distant relative of distance vector routing algorithm. It finds a route to a destination only when someone wants to send a packet to that destination.

Route Discovery : Two nodes are said to be connected, it they can communicate directly using their radios. Consider in the following adhoc network, soruce A wants to send a packet station I,



Suppose that AODV algorithm maintains a table at each node.

1. A looks in its routing table and does not find an entry for I
2. It now has to discover a route to I. This process of discovering routes is called on demand.

To discover I,

1. A constructs a special ROUTE REQUEST RRQ packet and broadcasts it
2. The RREQ packet reaches to B and D

RREQ packet consists of the following fields

Source Address	Request Id	Destination Address	Source seq no	Dest Seq no	Hop count

Request ID is a local counter, separately maintained by each node and incremented each time a RREQ is broadcasted. The sources address and Request id fields together identify a RREQ packet.

In addition Request ID, each node also maintain a second seq counter which is incremented whenever a RREQ is sent. Hop count will keep track of how many hops the packet has made. It is initialized to zero.

When an RRQ arrives at node B and B the following steps will happen

3. B and D verifies their routing table to check if they all received this request, with the same source address, and Request ID. If it is a duplicate, they discard the packet, otherwise, the data is entered into the routing table to avoid further duplicates and processing will continue
4. The receiver looks up the destination in its route table. If a fresh route to the destination is known, a Router Reply RREP packet is sent back to the source by giving information how to reach to the destination. If it is less, the route is older than the previous route the source had for the destination. Then it goes to the next step.
5. since the receiver does not know fresh route to the destination, it increments the hop count field and rebroadcast the RREQ packet. It also extracts the data from the packet and stores it as a new entry in its reverse route table.

→Neither B or D knows where I is, therefore each of them creates a reverse route entry pointing back to A and broadcasts the packet with hop count set to 1.

→ The broadcast from B reaches C and D.

→ C makes an entry for it in its reverse route table and rebroadcasts it.

→ D rejects it as a duplicate packet.

→ Similarly D's broadcast is rejected by B.

→ D's broadcast is accepted by F and G and stored.

→ After E,H and I receive the broadcast, the RREQ finally reaches to a destination that knows station I.

In response to the incoming request, station I builds a RREP packet. The contents of RREP packet are

Source address	Destination address	Destination seq no	Hop count	Lifetime

The source add, destination add and hop count will be copied form the incoming request. But the destination seq no taken from its counter in memory. The hop count is set to 0. the lifetime field controls how long the route is valid. This packet is unicasted to the node that the RREQ packet came form, here the node is G. It then follows the reverse path to D and finally to A. At each node, hop count is incremented so the node can see how far from the destination (I) is.

Congestion Control :

When too many packets are present in the subnet, performance degrades. This situation is called congestion. Congestion can be brought on several factors.

1. If all of a sudden, stream 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 or store all these packets, then packets will be lost. To recover

this problem, even if we add more memory to the routers, then also the packets life time may expire. Though we route the time out packets, destinations reject these packets.

2. slow processors can also cause congestion.

The difference between congestion control and flow control is, congestion control make sure that the subnet is able to carry the offered traffic. It is a global issue involving the behavior of all the hosts, and all routers.

Flow control in contrast, relates to the point to point traffic between a given sender and receiver. This job is to make sure that a fast sender can not continually transmit data faster than the receiver can absorb it.

Congestion control Algorithms:

Congestion control algorithms are designed for two kinds of systems open loop and closed loops systems.

Open loop systems decide when to accept new traffic and when to discard packets. Open loop algorithms work with source and destination.

Where as, closed loop systems based on the concept of a feedback loop. Closed loop algorithms are divided into two subcategories, explicit and implicit feed back. This approach has 3 steps

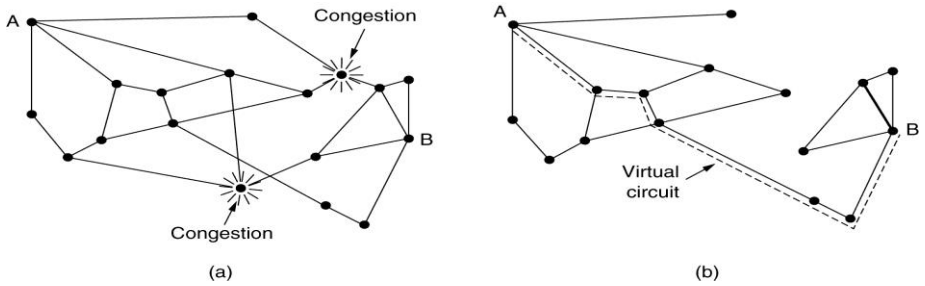
1. Monitor the system to detect when and where congestion occurs.
2. pass this information to places where action can be taken.
3. Adjust the system operation to correct the problem.

In explicit feedback, packets are sent back from the point of congestion to warn the source. In implicit algorithms, the source deduces the existence of congestion by making local observations, such as the time required to acks to come back.

Congestion Prevention Policies :

Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Congestion control in Virtual circuit subnets : One of the widely used technique to avoid congestion is admission control mechanism. Means once congestion has been signaled, no more virtual circuits are set up until the problem has gone away. An alternative approach is to allow new virtual circuits, but carefully route all new virtual circuits around problem areas.



Consider the above subnet, suppose that a host attached to router A, wants to set up a connection to host attached to router B. Normally, this connection would pass through one of the congested routers. To avoid this situation, we can redraw the subnet like in the 2nd diagram, by omitting the congested routers and all of their lines. The dashed line shows a possible route for the virtual circuit that avoids congested routes.

Congestion control in Datagram Subnets : Each router can easily monitor the utilization of its output lines and other resources. It can associate a value with each line, a real variable 'u' whose value is in between 0.0 and 1.0 reflects the recent utilization of the line.

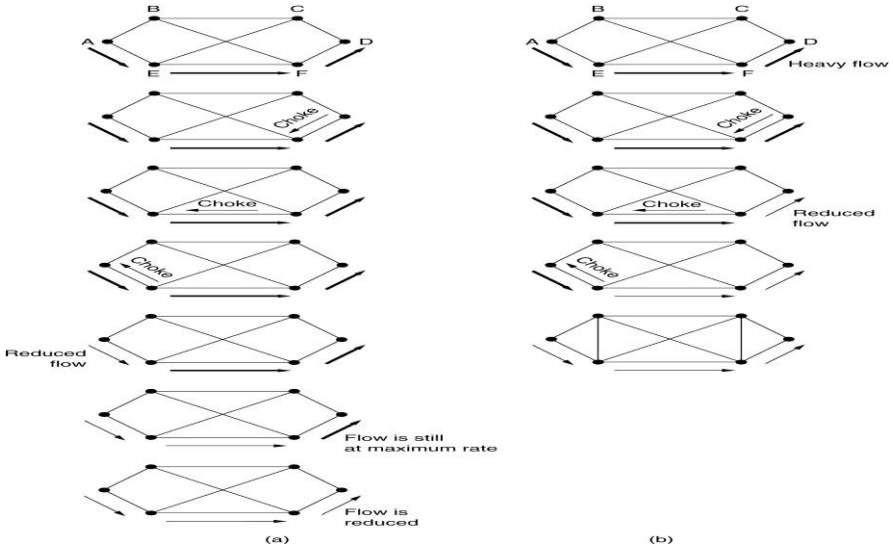
Whenever u moves above the threshold, the output line enters into a warning state. The newly arriving packet is checked to see if its output line is in warning state or not. If it is in warning state, an action is taken. The action can be taken based on the several alternatives.

1.Warning Bit : DECNET signals the warning state by setting a special bit in the packet's header. When the packet arrives at tits destination, the transport entity copied the bit into the next ack sent back to the source.

As long as the router was in the warning state, it continues to set the warning bit. This means that source continually gets acks with a warning bit set to 1. so source decrease its transmission rate.

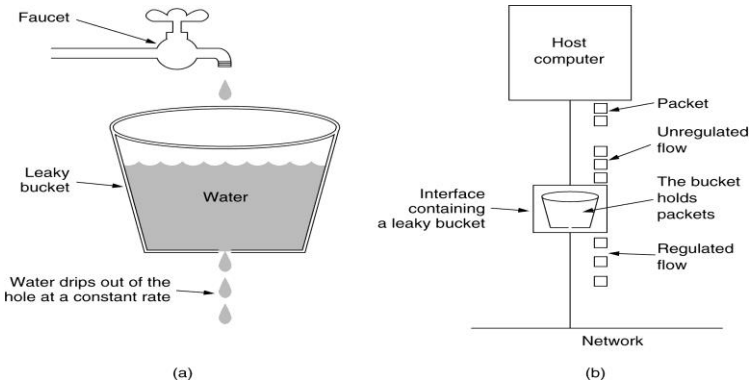
2. Choke packets : The previous congestion control algorithm is fairly clever. This algorithm uses a choke packet to warn the source station. The original packet is added to this choke packet and is forwarded to the destination. When the source host gets the choke packet, it is required to reduce the traffic sent to the specific destination by X percent. Hosts can reduce traffic by adjusting their policy parameters, for example, window size or leaky bucket output rate.

3. Hop By Hop Choke Packets : when we are using choke packets to control congestion, on long distance lines, it may take several seconds to reach to the original source. Within this much time source can generate many packets, all of them will be discarded due to traffic failures. That is why an alternative approach to have choke packet that take effect at every hop it passes through.



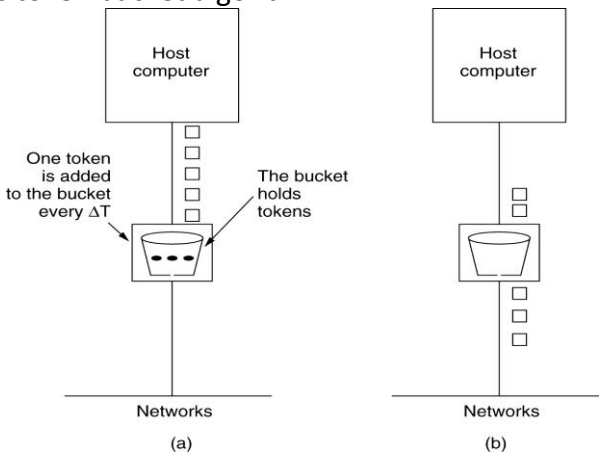
Here in the second diagram, as soon as the choke packet reaches F, F is required reduce the flow from F to D. Doing this, F can assign more buffers to flow. In the next step choke packet reaches to E, which tells E to reduce the flow to F. Finally choke packet reaches to A and flow will slow down. This helps us to get relief fastly from congested routers and traffic.

4. Leaky Bucket Algorithm : Imagine a bucket with a small whole in the bottom. It will not bother about the incoming data transmission rate, the outflow is at a constant rate, when there is any water in the bucket, and zero 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. If a packet arrives at the queue, when it is full the packet is discarded. In other wards, if one or more processes within the host try to send a packet is discarded. It was first proposed by Turner. In fact it is a single server queuing system with constant service time.



The host is allowed to put one packet per clock tick on to the network. Again this can be enforced by the interface card or by the operating system. This mechanism turns an uneven flow of packets from the user processes inside the host into an even flow of packets on to the network. Implementing the original leaky bucket algorithm is easy. The leaky bucket consists of a finite queue. When a packet arrives, if there is a room on the queue it is appended to the queue; otherwise it is discarded.

5. Token Bucket Algorithm : The leaky bucket algorithm enforces a rigid output pattern at the averaged rate, without considering how burst the traffic is. For many applications, it is better to allow the output to speed up somewhat when large bursts arrive. So a more flexible algorithm is needed, preferably one that never loses data. One such algorithm is token bucket algorithm.



In this algorithm, the leaky bucket holds tokens, generated by a clock a the rate of the token every second. Here in the above diagram, we see a bucket which holds 3 tokens, with 5 packets waiting to be transmitted. For a packet to be transmitted, it must capture and destroy one token. We see that 3 of 5 packets have passed through. But the other two are stuck waiting for two more tokens to be generated. Token bucket algorithm throws away token when the bucket is fills up but never discards packet.

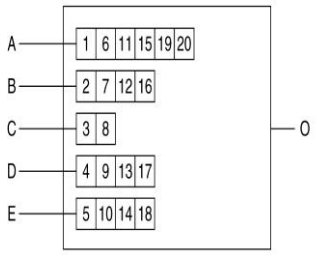
6. Flow Specifications : Traffic shaping is most effective when the sender, receiver and the subnet all agree to it. To get agreement it is necessary to specify the traffic pattern in a good way. Such an agreement is called a flow specification. It consists of data

structure that describes both the pattern of the injected traffic and the quality of service desired by the application. Flow specification can apply either to the packets sent on a virtual circuit, or to a sequence of data grams sent between source and destination.

Parameter	Unit
Token bucket rate	Bytes/sec
Token bucket size	Bytes
Peak data rate	Bytes/sec
Minimum packet size	Bytes
Maximum packet size	Bytes

7. Weighted fair Queuing : A problem with using choke packets is that the action to be taken by the source hosts is voluntary. Suppose that a router is being swapped by packets from four sources, and it sends choke packets to all four stations. One of them reduces its data transfer rate, but the other 3 just keep the same speed. The result is that the honest host gets an even smaller share of the bandwidth than it has before.

To get around this problem, Nagle proposed a fair queuing algorithm. The importance of the algorithm is routers have multiple queues for each output lien, one for each source. When a line becomes idle, the router scans the queues round robbin, taking the first packet on the queue.



Packet	Finishing time
C	8
B	16
D	17
E	18
A	20

(b)

In the above example, we have 2 to 6 bytes of information for each station. At first clock tick, first byte of the packet on line A is sent. Then first byte of the packet from line B and so on. The first station that finish, data transmission is C.

8. Load shedding : When none of the above methods help in avoiding congestion, then we use load shedding. It is a fancy way of saying that when routers are being inundated by packets that they can not handle, they just throw them away. This term comes from the word electrical power generation, where it refers to the practice of utilities intentionally blocking out certain areas to save the entire grid from collapsing on hot summer days when the demand for electricity greatly exceeds the supply.

A router drowning in packets can just pick packets at random to drop, but usually it can do better than that. The packet discarding policy will depend on the application. Suppose for file transfer, an old packet is more important than the new one. Always packet retransmission is necessary in these applications.

In contrast for multimedia, a new packet is more important than the old one. Here always late data is more worst than the bad data. Sometimes retransmission can cause serious problems.

9. Random Early detection : According to this, it discards packets before all the buffer space is really exhausted. Routers drop packets before the situation has become hopeless. To determine when to discard routers maintain a running average of their queue lengths. When the average queue length of some line exceeds a threshold, the line is said to be congested and the action is taken.

10. Jitter Control : For applications such as audio and video streaming, it is not a matter that packets take 20 msec or 30msec to deliver. The variation in the packet arrival time is called jitter. The jitter can be bounded by computing the expected transit time for each hop 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 and updated at each hop. If the packet is ahead of the schedule, it is held just long enough to get back on schedule. If it is behind schedule, the router tries to get it out quickly.

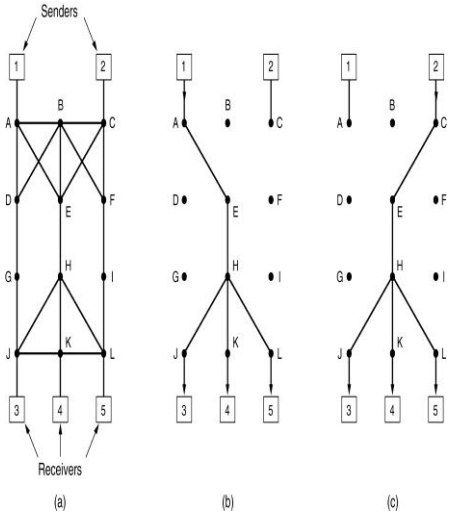
Quality of Service :

Requirements : A stream of packets from a source to a destination is called a flow. In a connection oriented network, all the packets belonging to a flow follow the same route. In a connectionless network, they may follow different routes.

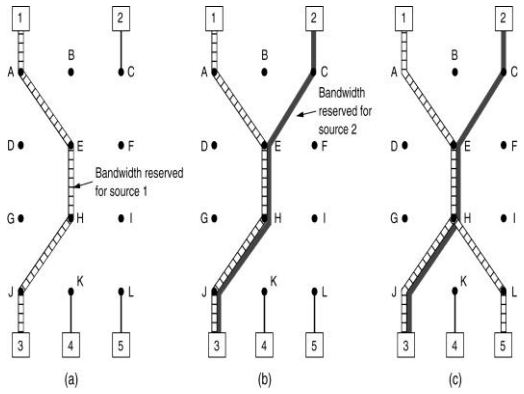
Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High	High

Congestion Control for multicasting :

Resource Reservation Protocol (RSVP) : It allows multiple senders to transmit to multiple groups or receivers. It permits individual receivers to switch channels freely, and optimizes bandwidth, and also eliminates congestion. Simple saying it uses multicast routing using spanning trees. Each group is assigned a group address. To send a packet to group, sender puts the group address in the packet. The algorithm then builds the spanning tree covering all group members.



Hosts 1 and 2 are multicast senders, and hosts 3,4, and 5 are multicast receivers. Here the senders and receivers are disjoint, but in general the two sets may overlap. The multicast trees for host1 and 2 are shown as separately. 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 information and reserves the necessary bandwidth. If sufficient bandwidth is available, it reports back failure.



An example of such a reservation is shown above. Here host 3 has requested a channel from channel 1. Once it has been established, packets can flow from 1 to 3 without congestion. Now consider, if host 3 next reserves the other channel 2 also, then the user can watch two programs at once. A second path is reserved like a second diagram. Two separate channels are needed host 3 to router E, because two independent streams are being transmitted. Same process happens for host 5 also.

Differentiated Services in Congestion Control :

Flow based algorithms have the potential to offer good quality of service to one or more flows because they reserve whatever resources are needed along the route. Differentiated services can be offered by a set of routers forming an administrative domain. The administrative defines a set of service classes with corresponding forwarding rules. These are represented in IP TOS field. It offers two kinds of services: that is flow based and class based services.

To make the difference between class based and flow based QOS, consider an example : Internet Telephony. With a flow based scheme, each telephone call gets its own resources and guarantees. With a class based scheme, all the telephone calls together gets the resources reserved for the class telephony. These resources can not be taken away by packets from the file transfer class or other classes, but no telephone call gets any private resources reserved for it alone.

Expedited Forwarding :

The choice of service classes if up to operator. The simplest service class is expedited forwarding. The term expedite means speed up and rush. The idea behind expedited forwarding is very simple. Two classes of service are available: regular and expedited. The vast majority of the traffic is expected to be regular, but a small fraction of the packets are expedited. The expedited packets should be able to transit the subnet as though no other packets were present.

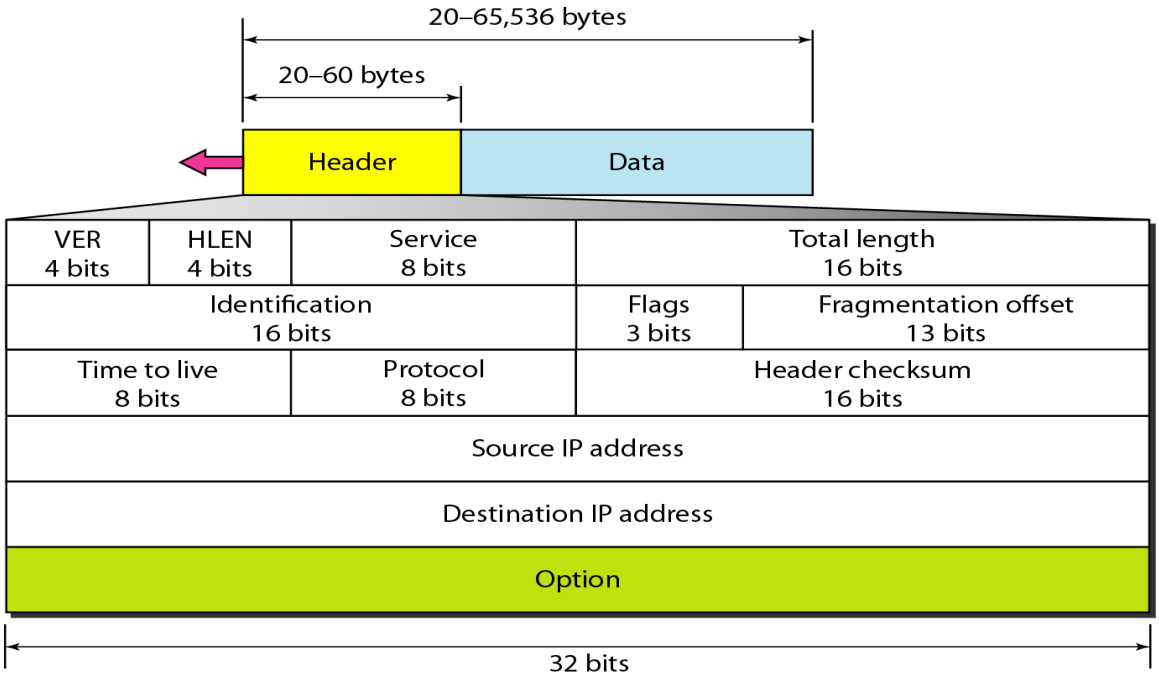
The two logical pipes shown in the figure represent a way to reserve bandwidth. One way to implement this strategy is to program routers to have two output queues for each outgoing line. One for expedited packets, and one for regular packets. When a

packets arrives it is queued accordingly. Packet scheduling use like weighted fair queuing. For example 10% of the traffic is expedited and 90% is regular. 20% of the bandwidth is dedicated to expedited and the rest to regular traffic. Therefore, expedited traffic will get twice the bandwidth of the regular. This allocation can be achieved by transmitting one expedited packet for every four regular packets.

Assured Forwarding : A more elaborate scheme to manage service classes if assured forwarding. There are four priority classes, each class has its own resources. It defines 3 discard probabilities for packets that are undergoing congestion: low, medium, and high.

1. Classify the packets into one of the four priority classes. It is done at the sending host or at the ingress(first) router.
2. Mark the packets according to their class. A header field is required for this. Fortunately an 8 bit service field is present in the IP Packet header.

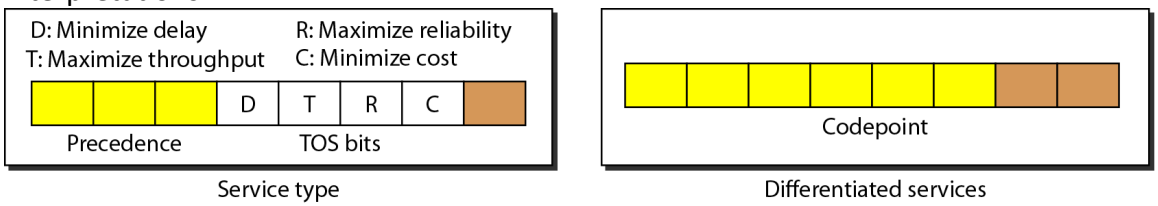
INTERNET PROTOCOL: IPV4 is the delivery mechanism used ny TCP/IP. IPV4 has 20 byte fixed header and 40 byte optional header. The total size of IP header is 60 bytes IPv4 is an unreliable and connectionless datagram protocol-a best-effort delivery service. The term best-effort means that IPv4 provides no error control or flow control (except for error detection on the header). IPv4 assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees. If reliability is important, IPv4 must be paired with a reliable protocol such as TCP. IPv4 is also a connectionless protocol for a packet-switching network that uses the datagram approach.



Version (VER): This 4-bit field defines the version of the IPv4 protocol. Currently the version is 4. However, version 6 (or IPng) may totally replace version 4 in the future.

Header length (HLEN): This 4-bit field defines the total length of the datagram header in 4-byte words. This field is needed because the length of the header is variable (between 20 and 60 bytes). When there are no options, the header length is 20 bytes, and the value of this field is 5 (5 x 4 = 20). When the option field is at its maximum size, the value of this field is 15 (15 x 4 = 60).

Services: IETF has changed the interpretation and name of this 8-bit field. This field, previously called service type, is now called differentiated services. We show both interpretations.



Default types of service

Protocol	*TOS Bits*	*Description*
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

Total Length: This is a 16-bit field that defines the total length (header plus data) of the IPv4 datagram in bytes. To find the length of the data coming from the upper layer, subtract the header length from the total length. The header length can be found by multiplying the value in the HLEN field by 4.

Length of data =total length - header length

Since the field length is 16 bits, the total length of the IPv4 datagram is limited to 65,535 (216 - 1) bytes, of which 20 to 60 bytes are the header and the rest is data from the upper layer.

Identification: This 16 bit field is used to identify a packet from source to destination. This is used in fragmentation.

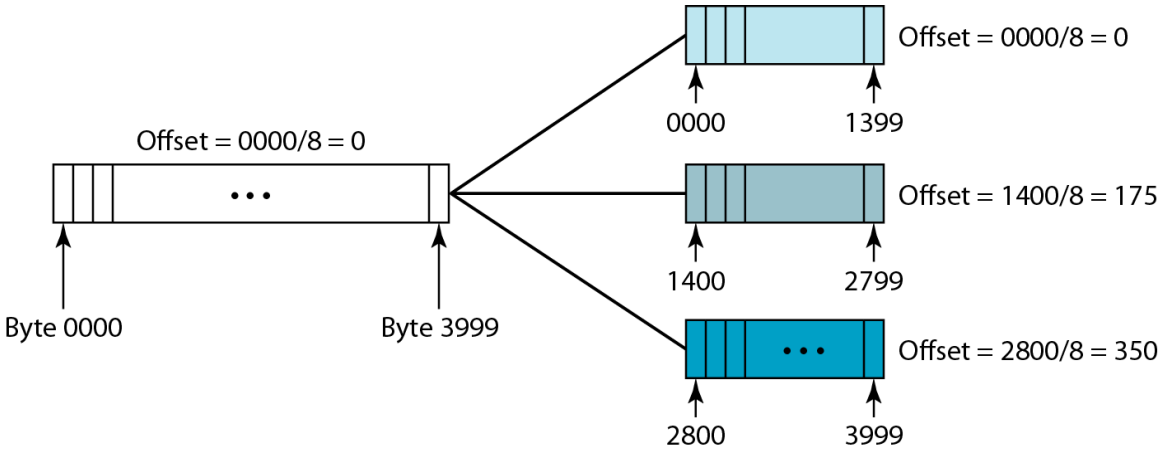
Flags: This is 3-bit field. The first bit is reserved.

The second bit is called do not fragment bit. If its value is 1, the machine must not fragment the packet. If it can not pass the packet through any network, it discards the packet and sends an ICMP error message to the source host. If its value is 0, the packet can be fragmented if necessary.

The third bit is called more fragment bit. If its value is 1, it means the packet not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

Fragmentation Offset: This 13 bit field shows the relative position of this fragment with respect to the whole packet.

Ex:



Time to Live: A packet has a limited life time in its travel through an internet. It was originally designed to hold a timestamp, which was decremented by each visited router. The packet was discarded when it reaches to zero. Today, this field is used mostly to control the maximum number of hops visited by the packet.

Protocol: This 8 bit field defines the higher level protocol that uses the services of the IPV4 layer.

Protocol values

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Checksum: 16 bit header checksum is used to detect he errors in IP headers.
Source address: This 32-bit field defines the IPv4 address of the source. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.
Destination address: This 32-bit field defines the IPv4 address of the destination. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.