Module 3

Network Layer

NETWORK LAYER

Ques 1) What is network layer? What are the functions of network layer?

Ans: Network Layer

The network layer is the third layer of OSI model. It responds to service requests from the transport layer and issues service requests to the data link layer.

Network layer addresses messages and translates logical addresses and names into physical addresses. It also determines the route from the source to the destination computer and manages traffic problems, such as switching, routing, and controlling the congestion of data packets.

Network layer describes how a series of exchanges over various data links can deliver data between any two nodes in a network. This layer defines the addressing and routing structure of the Internet. Network layer is concerned with controlling the operation of the subnet.

Functions of Network Layer

The specific functions of the network layer are given below:

1) **Logical Addressing:** The physical addressing implemented by the data link layer handles the addressing problem locally. If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems. The network layer adds a header to the packet coming from the upper layer that, among other things, includes the logical addresses of the sender and receiver.

2) **Routing:** When independent networks or links are connected together to create an Internetwork (a network of networks) or a large network, the connecting devices (called routers or gateways) route the packets to their final destination. One of the functions of the network layer is to provide this mechanism.

3) **Internetworking:** This is the main duty of network layer. It provides the logical connection between different types of networks.

4) **Packetising:** The network layer encapsulates the packets received from upper layer protocol and makes new packets. This is called as **packetising**. It is done by a network layer protocol called IP (Internetworking Protocol).

5) **Fragmenting:** The datagram can travel through different networks. Each router decapsulates the IP datagram from the received frame. Then the datagram is processed and encapsulated in another frame.

Ques 2) What are the different design issues of network layer?

Ans: Design Issues of Network Layer

The network layer design issues include:

1) **Services Provided to Transport Layer:** Main features of the services provided to transport layer are as follows:
   i) The services provided should be **independent of the underlying technology**. Users of the service need not be aware of the physical implementation of the network – for all they know, their messages could be transported *via* carrier pigeon! This design goal has great importance when one consider the great variety of networks in operation.

      In the area of Public networks, networks in underdeveloped countries are nowhere near the technological ability of those in the countries like the US or Ireland. The design of the layer must not disable from connecting to networks of different technologies.

   ii) The **transport layer (i.e., the host computer) should be shielded from the number, type and different topologies of the subnets uses**. That is, all the transport layer wants is a communication link; it need not know how that link is made.

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

2) **Internal Design of Subnet:** There are basically two different philosophies for organising the subnet:
   i) **Connections:** In the context of the internal operation of the subnet, a connection is usually called a **virtual circuit**.
   ii) **Connectionless:** The independent packets of the connectionless organisation are called **datagrams**.

Ques 3) What is routing? What are the design goals of routing algorithm? List out the different types of routing algorithms.

Ans: Routing

Routing is the process of selecting paths in a network along which to send network traffic. Routing is usually performed by a dedicated device called a router.

Scanned by CamScanner

A router is a networking device that forwards packets between networks using information in protocol headers and forwarding tables to determine the best next router for each packet. Routers work at the Network Layer (layer 3) of the OSI model and the Internet Layer of TCP/IP.

For routing of packets, routing algorithms are used. The routing algorithm is that part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on.

Routing algorithms can be differentiated based on several key characteristics:

1) First, the particular goals of the algorithm designer affect the operation of the resulting routing protocol.
2) Second, various types of routing algorithms exist, and each algorithm has a different impact on network and router resources.
3) Finally, routing algorithms use a variety of metrics that affect calculation of optimal routes.

Design Goals of Routing Algorithms

1) **Optimality:** Optimality refers to the capability of the routing algorithm to select the best route, which depends on the metrics and metric weightings used to make the calculation.

   For example, one routing algorithm may use a number of hops and delays, but it may weigh delay more heavily in the calculation. Naturally, routing protocols must define their metric calculation algorithms strictly.

2) **Simplicity and Low Overhead:** Routing algorithms also are designed to be as simple as possible. In other words, the routing algorithm must offer its functionality efficiently, with a minimum of software and utilisation overhead. Efficiency is particularly important when the software implementing the routing algorithm must run on a computer with limited physical resources.

3) **Robustness and Stability:** Routing algorithms must be robust, which means that they should perform correctly in the face of unusual or unforeseen circumstances, such as hardware failures, high load conditions, and incorrect implementations. Because routers are located at network junction points, they can cause considerable problems when they fail. The best routing algorithms are often those that have withstood the test of time and that have proven stable under a variety of network conditions

4) **Rapid Convergence:** In addition, routing algorithms must converge rapidly. **Convergence** is the process of agreement, by all routers, on optimal routes. When a network event causes routes to either go down or become available, routers distribute routing update messages that permeate networks, stimulating recalculation of optimal routes and eventually causing all routers to agree on these routes. Routing algorithms that converge slowly can cause routing loops or network outages.

5) **Flexibility:** Routing algorithms should also be flexible, which means that they should quickly and accurately adapt to a variety of network circumstances. Assume, **for example,** that a network segment has gone down. As many routing algorithms become aware of the problem, they will quickly select the next-best path for all routes normally using that segment. Routing algorithms can be programmed to adapt to changes in network bandwidth, router queue size, and network delay, among other variables.

Types of Routing Algorithm

Internet routing protocols employ one of following algorithms to gathering and using routing information:

1) Shortest Path Routing
2) Link-State Routing
3) Distance-Vector Routing
4) Flood-Based Routing Algorithm

Ques 4) Define the routing table.

Ans: Routing Table

Routing table is an electronic document that stores routes to various nodes in a computer network. The nodes may be any kind of electronic device connected to the network. The routing table is usually stored in a router or networked computer in the form of a database or file.

Network id	Cost	Next hop
.....	.....	.....
.....	.....	.....

Figure 3.1: Format of Routing Table

When data needs to be sent from one node to another on the network, the routing table is referred to in order to find the best possible route for the transfer of information.

The routing table consists of at least three information fields:

1) **Network ID:** i.e., the destination network id.
2) **Cost:** i.e., the cost or metric of the path through which the packet is to be sent.
3) **Next Hop:** The next hop, or gateway, is the address of the next station to which the packet is to be sent on the way to its final destination.

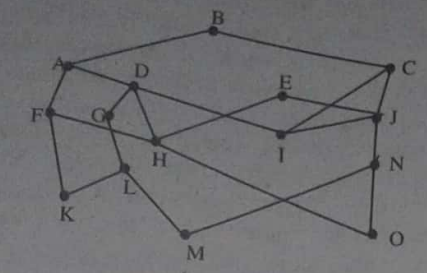
Ques 5) Explain the Optimality Principle.

Ans: Optimality Principle

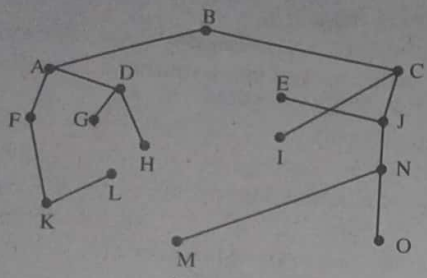
Optimality principle states that if router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route.

To see this, call the part of the route from I to J r_1 and the rest of the route r_2. If a route better than r_2 existed from J to K, it could be concatenated with r_1 to improve the route from I to K, contradicting our statement that r_1r_2 is optimal.

Scanned by CamScanner



(a)



(b)

Figure 3.2: (a) A Subnet. (b) A Sink Tree for Router B

As a direct consequence of the optimality principle, we can see that the set of optimal routes from all sources to a given destination form a tree rooted at the destination. Such a tree is called a sink tree and is illustrated in **figure 5.1,** where the distance metric is the number of hop. Sink tree is not necessarily unique; other trees with the same path lengths may exist. The goal of all routing algorithms is to discover and use the sink trees for all routers.

Ques 6) Discuss the shortest path routing. Also explain the Dijkstra's Algorithm in detail.

Ans: Shortest Path Routing

Shortest Path Routing is suited for static routing. A path selected can be called shortest in many contexts. If one selects cost as criteria then the shortest path is the route which is least expensive. If distance is the criteria for determining shortest path then minimum length path is taken in to consideration.

If time is the criteria then the path which takes least time to reach the destination is called shortest path. One can use anyone of the following shortest path routing algorithms:

Dijkstra Algorithm

In this algorithm the criteria for shortest path is distance. All distances being known, in this method the shortest path with respect to distance is looked for from source to destination. This is also called minimum cost or **Least cost algorithm.**

The vertices are assumed to act as routers and edges act as connecting media. Dijkstra's algorithm is used to find the shortest path between any two vertices s and t in G.

The principle behind Dijkstra's algorithm is that if s, . . . x, . . . t is the shortest path from s to t, then s, . . . x had better be the shortest path from s to x.

This suggests a dynamic programming-like strategy, where one store the distance from s to all nearby nodes, and use them to find the shortest path to more distant nodes.

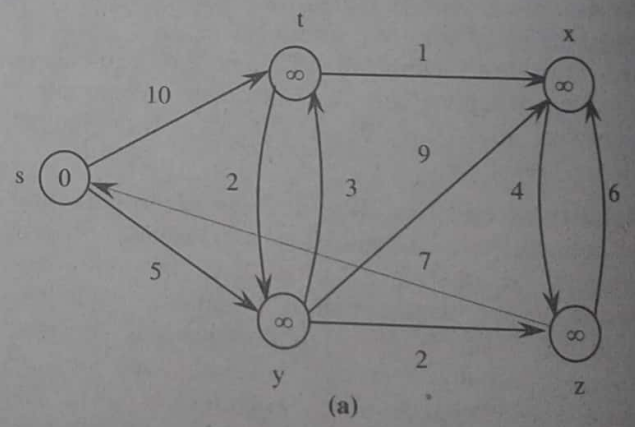
The shortest path from s to s, d(s, s) = 0. If all edge weights are positive, the smallest edge incident to s, say (s, x), defines d(s, x).

An array is used to store the length of the shortest path to each node. Initialize each to 1 to start. Soon as the shortest path is established from s to a new node x, go through each of its incident edges to see if there is a better way from s to other nodes through x.

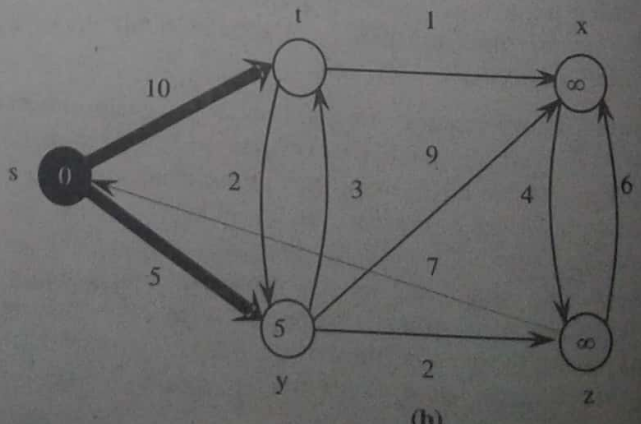
Steps: Dijkstra Algorithm

known = {s}
for i = 1 to n, dist[i] =∞
for each edge (s, v), dist[v] = d(s, v)
last=s
while (last ≠ t)
select v such that dist(v) = $\min_{unknown}$ dist(i)
for each (v, x), dist[x] = min(dist[x], dist[v]+ w(v, x))
last = v
known = known U {v}

For example, consider **figure 3.3** in which source s is the leftmost vertex. The shortest-path estimates are shown within the vertices, and shaded edges indicate predecessor values. Black vertices are in the set S, and white vertices are in the min-priority queue Q = V – S.

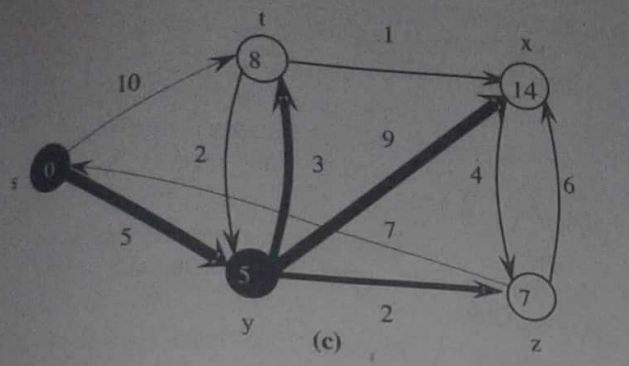


(a)

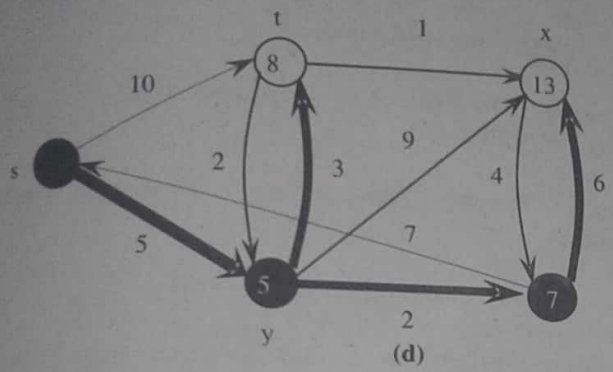


(b)

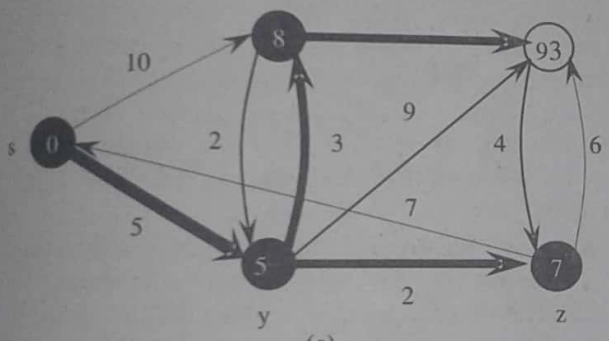
Scanned by CamScanner



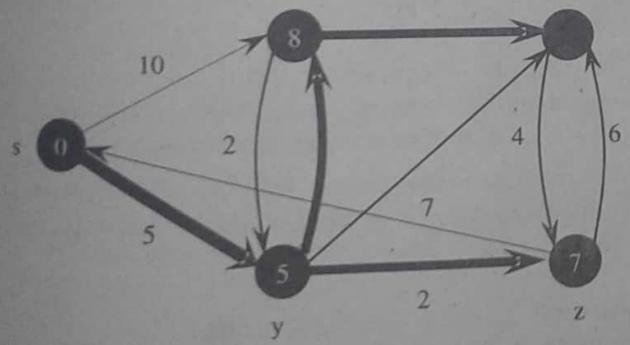
(c)



(d)



(e)



(f)

Figure 3.3: Execution of Dijkstra's Algorithm

In **Figure 3.3(a)** the situation just before the first iteration of the while loop of lines 4-8. The shaded vertex has the minimum d value and is chosen as vertex u in line 5. In **Figure 3.3(b)-(f)** the situations after each successive iteration of the while loop are shown. The shaded vertex in each part is chosen as vertex u in line 5 of the next iteration. The d value shown in part (f) are the final values.

So the shortest path from s to t is 8, s to y is 5, s to z is 7, s to x is 9

Ques 7) Discuss the Bellman-Ford Algorithm with suitable example.

Ans: Bellman-Ford Algorithm

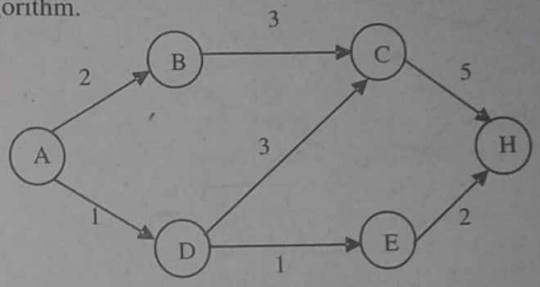
This algorithm is suitable for a directed graph. In this case too the least cost distance from every node in a network to a special node is found out. If for a given graph it is required to find the minimum path from all nodes to A then we proceed in such a way that we consider all those nodes which can reach that particular node in a single hop.

Each node is marked in the format,

$$D_x^y = d$$

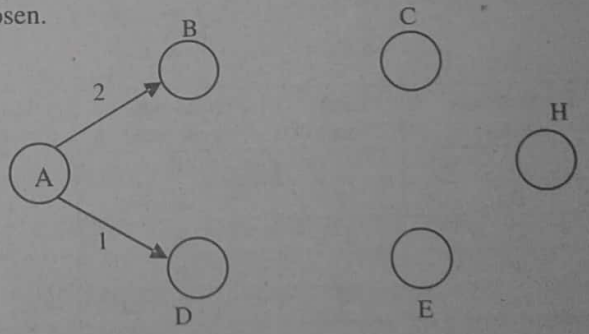
Where x is the node number, y the hops considered and d is the distance. If there are nodes which are not directly connected we mark their d as infinity. We continue evaluating this distance Vs hops till we have considered hop value upto one less than the number of nodes.

Example: Consider the following graph find the shortest path between node A and node H using Bellman-Ford algorithm.



Solution:

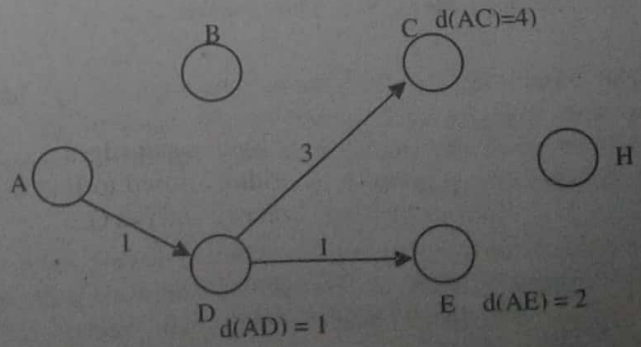
Step 1: Distance AD is shorter than AB. So route AD is chosen.



Step 2:

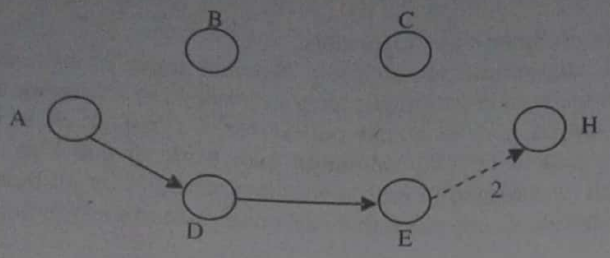
$\because$ d(AE) < d(AC)

$\therefore$ d(AE) is chosen.



Scanned by CamScanner

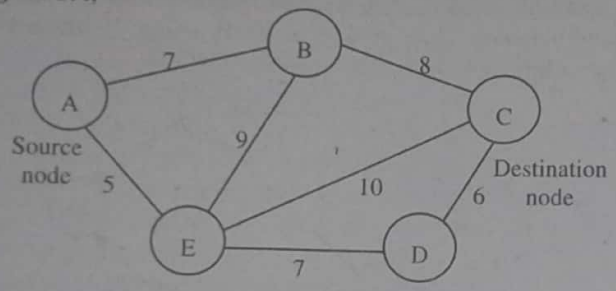
Step 3: So the shortest distance is ADEH, the result is same as in Dijkstra's algorithm.



Ques 8) Explain link state routing algorithm.

Ans: Link State Routing Algorithm

In this algorithm, exchange of the link-state packets over the subnet hold key to facilitating the routing process. In this algorithm, network topology and link costs are estimated by making each node broadcast what is referred to as 'link-state packets' carrying 'identities of neighbours' and 'corresponding link costs', as shown in **figure 3.4**.



A	
11...001	
60	
B	7
E	5

Destination (router)	Link-cost	Next hop (router)	Hop count
A	0	A	1
B	7	B	1
C	15	B	2
D	12	E	2
E	5	E	1

Source	Sequence No.	Age	Send Flags	Acknowledgement Flags	Data

Figure 3.4: Structure of a Link-State Packet, Routing Table and Packet Buffer (at router A)

The basic idea involves the computation of the local routing table by each router on the basis of its own estimates and the similar link-state broadcasts received from other routers in the subnet.

In a simple version each router following this algorithm:

1) Discovers its neighbours and their network addresses by sending special packets called 'hello' packets,
2) Estimates delay/cost or any other metric for reaching its neighbours by sending another special packet called 'echo' packets.
3) Immediately applies its recent knowledge to form link-state packet, which encapsulates this estimate and, sends (broadcasts) the packet to all the discovered routers.
4) Computes the shortest path to every other router using the shortest path algorithm and updates the local routing table.
5) Immediately forms fresh Link-State Packets (LSPs) and executes link-state broadcast. This is sometimes called **controlled flooding**.

Ques 9) Describe the Distance Vector Routing Algorithm in detail.

Or

Discuss the problems occurred in distance vector routing.

Ans: Distance Vector Routing Algorithm

Distance Vector Routing (DVR) is also known as the **Bellman-Ford or Ford-Fulkerson routing algorithm.**

It is, the original dynamic routing algorithm used in the erstwhile ARPANET.

This scheme may be expressed as:

1) Each router knows/discovers its distance from its neighbours,
2) Each router locally maintains a routing table indexed by an entry for every other router in the subnet and identification of a preferred neighbour/link leading to that router,
3) Metric of estimation may vary. **For example,** it may be any one of physical distance, hops, delay, etc.,
4) Periodically, each router sends a vector to its neighbouring routers. As this vector contains estimated distances, it is called a distance vector.
5) On receipt of such vectors from its neighbours, every router revises its estimates and updates its local routing table.

Problems in Distance Vector Routing

The problem which arises in the Distance Vector Algorithms is given below:

1) **Count-to-Infinity Problem:** Consider a router whose best route to destination X is large. If on the next exchange neighbour A suddenly reports a short delay to X, the router just switches over to using the line to A to send traffic to X.

   In one vector exchange, the good news is processed. Consider the five-node (linear) subnet of **figure 3.5**, where the delay metric is the number of hops. Suppose A is down initially and all the other routers know this. In other words, they have all recorded the delay to A as infinity.

Scanned by CamScanner

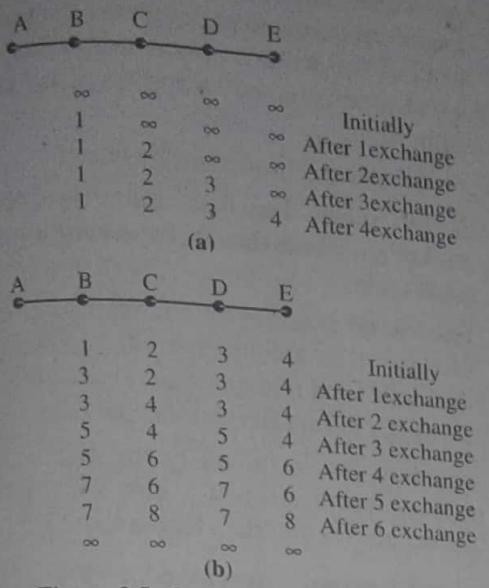


Figure 3.5: Count-to-infinity Problem

When A comes up, the other routers learn about it via the vector exchanges. For simplicity assume that there is (very big warning bell) somewhere that is struck periodically to initiate a vector exchange at all routers simultaneously. At the time of the first exchange, B learns that its left neighbour has zero delay to A. B now makes an entry in its routing table that A is one hop away to the left. All the other routers still think that A is down. At this point, the routing table entries for A are as shown in the second row of the **figure 3.5 (a)**.

On the next exchange, C learns that B has a path of length 1 to A, so it updates its routing table to indicate a path of length 2, but D and E do not hear the good news until later. Clearly, the good news is spreading at the rate of one hop per exchange. In a subnet whose longest path is of length N hops, within N exchanges everyone will know about newly revived lines and routers.

Consider the situation of **figure 3.5 (b)** in which all the lines and routers are initially up. Routers B, C, D, and E have distances to A of 1, 2, 3, and 4, respectively. Suddenly A goes down, or alternatively, the line between A and B is cut, which is effectively the same thing from B's point of view.

At the first packet exchange, B does not hear anything from A. Fortunately, C says "Do not worry. I have a path to A of length 2." Little does B know that C's path runs through B itself? For all B knows, C might have ten outgoing lines all with independent paths to A of length 2. As a result, B now thinks it can reach A via C, with a path length of 3. D and E do not update their entries for A on the first exchange.

On the second exchange, C notices that each of its neighbours claims to have a path to A of length 3. It picks one of them at random and makes its new distance to A 4, as shown in the third row of **figure 3.5 (b)**. Subsequent exchanges produce the history shown in the rest of **figure 3.5 (b)**.

From this figure, it is clear no router ever has a value more than one higher than the minimum of all its neighbours. Gradually, all the routers work their way up to infinity, but the number of exchanges required depends on the numerical value used for infinity. For this reason, it is wise to set infinity to the longest path plus 1. If the metric is time delay, there is no well-defined upper bound, so a high value is needed to prevent a path with a long delay from being treated as down. This problem is known as the count-to-infinity problem.

2) **Split Horizon Hack:** The split horizon algorithm works the same way as distance vector routing, except that the distance to X is not reported on the line that packets for X are sent on (actually, it is reported as infinity).

In the initial state of above **figure 3.5 (b), for example,** C tells D the truth about the distance to A, but C tells B that its distance to A is infinite. Similarly, D tells the truth to E but lies to C.

On the first exchange, B discovers that the direct line is gone, and C is reporting an infinite distance to A as well. Since neither of its neighbours can get to A, B sets its distance to infinity as well. On the next exchange, C hears that A is unreachable from both of its neighbours, so it marks A as unreachable too.

Using split horizon, the bad news propagates one hop per exchange. This rate is much better than without split horizon. The split horizon, although widely used, sometimes fails. Consider, **for example,** the four-node subnet of **Figure 3.6**. Initially, both A and B have a distance 2 to D, and C has a distance 1 there.

Now suppose that the CD line goes down. Using split horizon, both A and B tell C that they cannot get to D. Thus C immediately concludes that D is unreachable and reports this to both A and B.

Unfortunately, A hears that B has a path of length 2 to D, so it assumes it can get to D *via* B in 3 hops. Similarly, B concludes it can get to D via A in 3 hops. On the next exchange, they each set their distance to D to 4. Both of them gradually count to infinity, precisely the behaviour we were trying to avoid.

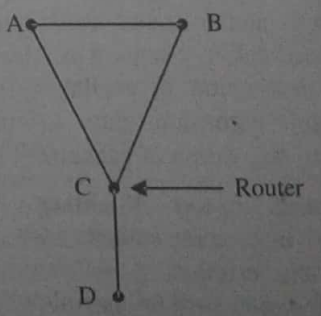


Figure 3.6: An Example where Split Horizon Fails

Scanned by CamScanner

Ques 10) What do you understand by flooding?

Or

Explain the flood-based routing algorithm.

Ans: Flood-based Routing Algorithm/Flooding

Flooding adapts the technique in which every incoming packet is sent on every outgoing line except the one on which it arrived. One problem with this method is that packets may go in a loop.

As a result of this a node may receive several copies of a particular packet which is undesirable. Some techniques adapted to overcome these problems are as follows:

1) **Sequence Numbers:** Every packet is given a sequence number. When a node receives the packet it sees its source address and sequence number. If the node finds that it has sent the same packet earlier then it will not transmit the packet and will just discard it.
2) **Hop Count:** Every packet has a hop count associated with it. This is decremented (or incremented) by one by each node which sees it. When the hop count becomes zero (or a maximum possible value) the packet is dropped.
3) **Spanning Tree:** The packet is sent only on those links that lead to the destination by constructing a spanning tree routed at the source. This avoids loops in transmission but is possible only when all the intermediate nodes have knowledge of the network topology.

Flooding is not practical for general kinds of applications. But in cases where high degree of robustness is desired such as in military applications, flooding is of great help.

Flood-based routing, as the name suggests, uses redundant replication of incoming packets/NLDUs on available outgoing links.

Variants of Flood-based Routing

1) **Pure Flooding Algorithm:** This is one of the simplest algorithms available to date that has a simple logic that suggests that if a packet arrives at a node that is member of the flood-based routing architecture, simply copy it (by replicating the original) on all outgoing links other than the link going back to the node wherefrom the packet has just arrived.

   Although under extreme unpredictability, this algorithm demonstrates consistent robustness and guaranteed delivery as long as atleast one path leading to the destination is available, it is inherently an inefficient algorithm due to the possibility of indefinite circulation of packets/NLDUs.

2) **Hop-Count based Flooding Algorithm:** This algorithm may be expressed as follows:
   i) At any originating node's', structure a packet such that its header contains a 'hop-count' that be initialised to length of the path (if known) or full diameter of the subnet.
   ii) At every intermediate node 'i', examine the incoming queue of packets, take the packet at the head of the queue and note the packet-id, line on which it arrived on, its hop count and destination address.
   iii) Decrement the hop-count by one '1'.
   iv) If the count becomes zero, discard/drop the packet and flush the corresponding entries in the local table.
   v) Otherwise, generate (n – 1) replicas of the packet (where 'n' is the number of arcs converging at this node) and transmit one replica on each of the arcs/lines except the one this packet arrived on.
   vi) Examine the incoming queue and if it is non-empty, repeat steps 2-5 else wait until a new packet arrives and then repeat steps 2-5.
3) **Selective/Direction-constrained Flooding Algorithm:** It is a variant of the basic flooding algorithm with the constraint of direction thrown in for the purpose of improved efficiency. In this scheme, packets are selectively flooded by the routers in such a way that they move approximately in the right direction (i.e., leading towards the destination).

Ques 11) What do you understand by routing protocols? What are the types of routing protocols?

Or

Explain the following protocols:

1) Interior Gateway Protocols (IGPs)
2) Exterior Gateway Protocols (EGPs)

Ans: Routing Protocols

A routing protocol specifies how routers communicate with each other, disseminating information that enables them to select routes between any two nodes on a computer network. Routing algorithms determine the specific choice of route. Each router has a priori knowledge only of networks attached to it directly. A routing protocol shares this information first among immediate neighbors, and then throughout the network. This way, routers gain knowledge of the topology of the network.

Types of Routing Protocol

Internet routing can be defined more precisely. All Internets routing protocols fall into one of the two categories:

1) **Interior Gateway Protocols (IGPs):** The router within an autonomous system uses an Interior Gateway Protocol (IGP) to exchange routing information. There are several IGPs available; each autonomous system is free to choose its own IGP. Usually, an IGP is easy to install and operate, but an IGP may limit the size or routing complexity of an autonomous system.

   There are two types Interior Gateway Protocols:
   1) **Interior gateway protocols type 1**, link-state routing protocols, such as Open Shortest Path First (OSPF) and IS-IS.

Scanned by CamScanner

2) **Interior gateway protocols type 2**, distance-vector routing protocols, such as Routing Information Protocol, RIPv2, IGRP. **Enhanced Interior Gateway Routing Protocol (EIGRP)** is an advanced distance-vector routing protocol that is used on a computer network for automating routing decisions and configuration.

2) **Exterior Gateway Protocols (EGPs):** A router is one autonomous system uses an exterior gateway Protocol (EGP) to exchange routing information with a router in another autonomous system. EGPs are usually more complex to install and operate than IGPs, but EGPs offer more flexibility and lower overhead (i.e., less traffic). To save traffic, an EGP summarises routing information from the autonomous system before passing it to another autonomous system. More important, an EGP implements policy constraint that allows a system manager to determine exactly what information is released outside the organisation.

Exterior gateway protocols are routing protocols used on the Internet for exchanging routing information between Autonomous Systems, such as Border Gateway Protocol (BGP), Path Vector Routing Protocol.

Figure 3.7 illustrates the two-level routing hierarchy used in the global internet:

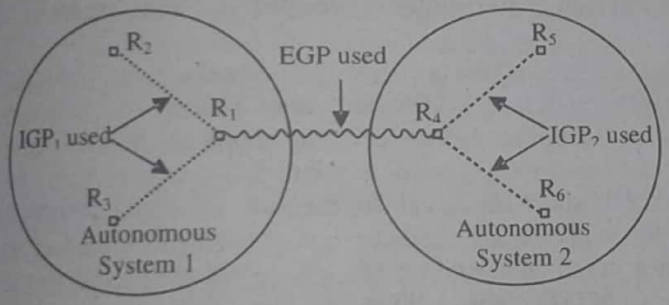


Figure 3.7: Internet Routing Architecture. Each Autonomous System Chooses an IGP to Use Internally; an EGP is used to Communicate between Autonomous Systems

In the **figure 3.7**, Autonomous System 1 (AS_1) has chosen IGP_1 to use internally, and Autonomous System 2 (AS_2) has chosen IGP_2. All routers in AS_1 communicate using IGP_1, and all routers in AS_2 communicate using IGP_2, Routers R_1 and R_4 use an EGP to communicate between the two autonomous systems. That is, R_1 must summarise information from its autonomous system and send the summary to R_4. In addition, R_1 accepts a summary from R_4, and uses IGP_1 to propagate the information to routers in AS_1 R_4 performs the same service for AS_2.

Ques 12) Discuss about RIP and OSPF?

Ans: Routing Information Protocol (RIP)

The Routing Information Protocol (RIP) is one of the oldest distance-vector routing protocols which employ the hop count as a routing metric. RIP prevents routing loops by implementing limit on the number of hops allowed in a path from source to destination. The maximum number of hops allowed for RIP is 15, which limits the size of networks that RIP can support. A hop count of 16 is considered an infinite distance and the route is considered unreachable. RIP implements the split horizon, route poisoning and hold down mechanisms to prevent incorrect routing information from being propagated.

Originally, each RIP router transmitted full updates every 30 seconds. In the early deployments, routing tables were small enough that the traffic was not significant. As networks grew in size, however, it became evident there could be a massive traffic burst every 30 seconds, even if the routers had been initialized at random times. It was thought, as a result of random initialisation, the routing updates would spread out in time, but this was not true in practice.

Sally Floyd and Van Jacobson showed in 1994 that, without slight randomisation of the update timer, the timers synchronised over time.

In most networking environments, RIP is not the preferred choice for routing as its time to converge and scalability are poor compared to EIGRP, OSPF, or IS-IS. However, it is easy to configure, because RIP does not require any parameters unlike other protocols.

RIP uses the User Datagram Protocol (UDP) as its transport protocol, and is assigned the reserved port number 520.

Open Shortest Path First (OSPF)

OSPF stands for Open Shortest Path First which uses link-state routing algorithm. Using the link state information which is available in routers, it constructs the topology in which the topology determines the routing table for routing decisions. It supports both variable-length subnet masking and classless inter-domain routing addressing models.

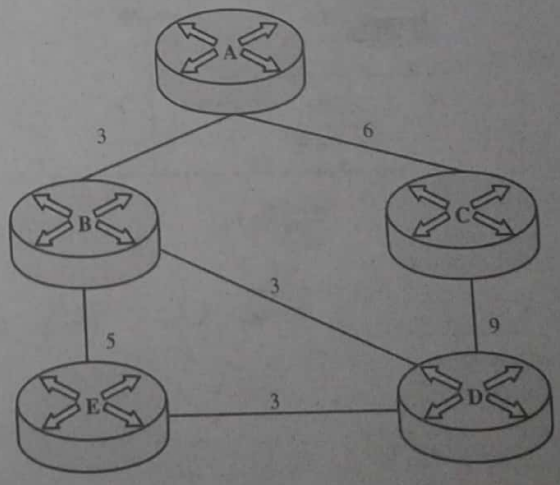


Figure 3.8: Simple Structure of OSPF

Since, it uses Dijkstra's algorithm, it computes the shortest path tree for each route. The main advantages of the OSPF (Open Shortest Path first) is that it handles the error detection by itself and it uses multicast addressing for routing in a broadcast domain.

Scanned by CamScanner

SPF Calculation

Before running the calculation, it is required that all routers in the network to know about all the other routers in the same network and the links among them. The next step is to calculate the shortest path between each single router. For all the routers they exchange link-states which would be stored in the link-state database. Every time a router receives a link-state update, the information stores into the database and this router propagate the updated information to all the other routers. Below is a simple model of how the SPF algorithm works.

A simple network formed by five routers; all the routers know about all the other routers and links. After all the paths are figured out, the path information are stored in the link database. The link database for the above model is : [A, B, 3], [A, C, 6], [B, A, 3], [B, D, 3], [B, E, 5], [C, A, 6], [C, D, 9], [D, C, 9], [D, B, 3], [D, E, 3] , [E, B, 5] and [E, D, 3].

Each term is referred to the originating router, the router connected to and the cost of the link between the two routers. Once the database of each router is finished, the router determines the Shortest Path Tree to all the destinations. (The shortest path in the SPF algorithm is called the Shortest Path Tree).

The Dijkstra's Shortest Path First is then running to determine the shortest path from a specific router to all the other routers in the network. Each router is put at the root of the Shortest Path Tree and then the shortest path to each destination is calculated. The accumulated cost to reach the destination would be the shortest path.

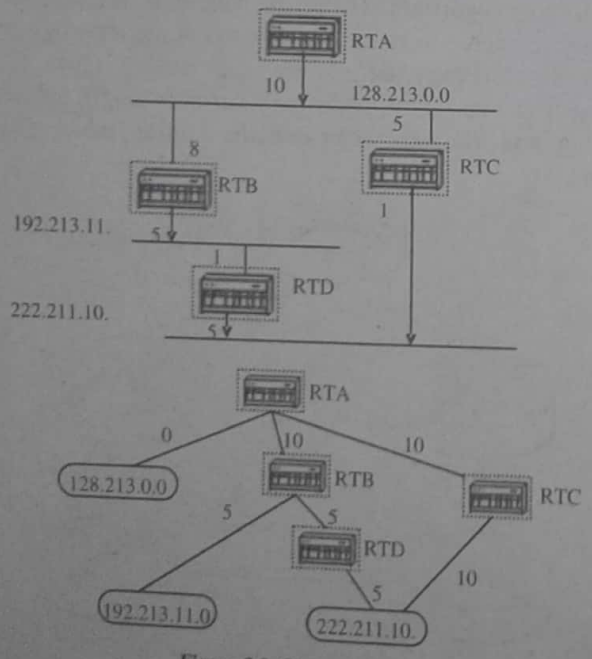


Figure 3.9: Shortest Path Tree

The cost (metric) of OSPF is the cost of sending packets across a certain interface. The formula to calcite the cost is:

cost = 10000 0000 /bandwidth in bps.

If the bandwidth is wider, the cost would be lower.

Above is a **figure 3.9** of the structure used to calculate the Shortest Path Tree. When the Shortest Path Tree is completed, the router will work on the routing table.

Ques 13) Explain the Routing for Mobile Hosts.

Or

What is mobile routing in the telephone network?

Ans: Routing for Mobile Hosts

What happens if a destination is not attached by a wire to a router, but instead can move about? Packets destined to that host somehow have to be forwarded to its new location, wherever it may be. The problem naturally resolves itself into two parts:

1) Finding-out where a host is, and
2) Getting packets or calls to it.

Mobile Routing in the Telephone Network

Cellular telephones use radio frequencies to communicate with a base station – usually located on a tall tower with a triangular platform on top, which you can see along major highways or in city centers – that relays their call to a Mobile Telephone Switching Office (MTSO). (To prevent unfair advantage to the local telephone company, the Federal Communications Commission in the United States requires that MTSOs be separated from central offices, though they serve nearly the same purpose.) Routing calls to and from a cellular telephone that may be associated with any MTSO in the cellular-service provider's service area.

Each cellular phone is statically assigned a globally unique ID and a home MTSO that does billing and provides access to the long-distance (toll) telephone network. The phone is also assigned a telephone number from the address space assigned to the home MTSO. When a cellular phone is switched on, it uses ALOHA contention on a common signaling channel to identify itself to the local MTSO. The MTSO, in turn, contacts the home MTSO and informs it of the phone's location.

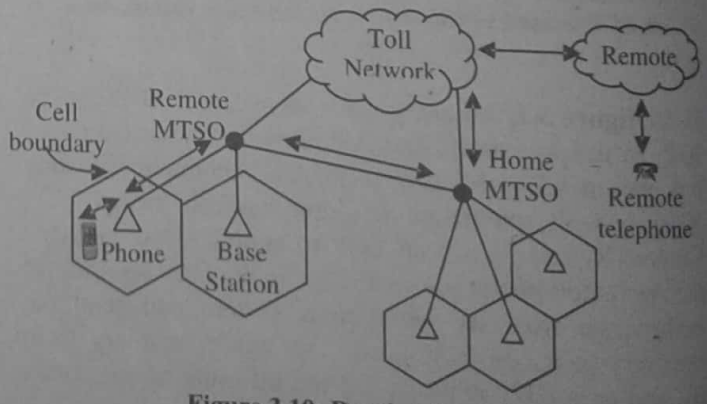


Figure 3.10: Routing for Cellular Phones.

When someone makes a call to the phone, the telephone network delivers it automatically to the home MTSO, which sets-up a connection to the phone through the remote MTSO, using Signaling System 7(SS7) signaling (**figure 3.10**). The remote MTSO contacts the nearest base station, which rings the cellular phone. The identity of the nearest base station is dynamically updated using the cellular hand-off.

Scanned by CamScanner

In the **figure 3.10**, Each Cellular Phone is assigned to a Mobile Telephone Switching Office (MTSO). Calls to the Phone are Routed through the Home MTSO to the Nearest Base Station via a Remote MTSO. As the Phone Moves, the Home MTSO is updated with the Location of the MTSO nearest the Phone.

To keep billing and accounting simple, all calls from the phone are always routed back to the home MTSO before they enter the toll network. Thus, the remote MTSO acts like a dumb switch to route calls to and from the home MTSO. If the phone moves from one MTSO to another, this information is sent back to the home MTSO, which updates its local database. Calls in progress are re-routed from the remote MTSO to the home MTSO, again using SS7 signaling.

This architecture allows cellular phones to roam within the entire service area freely, but has the overhead that calls are always routed to the home MTSO, requiring additional hops in the network. It scales well, because new MTSOs can be added dynamically as service demands increase. Moreover, heavily loaded MTSOs can be split to share the load. The cellular phone solution to mobility is simple and robust and has been modified for solving the mobility problem on the Internet.

Ques 14) What is mobile Routing in the Internet?

Ans: Mobile Routing in the Internet

Extensions to the standard solution that add robustness, efficiency, and security are still areas of active research. The field has evolved its own set of acronyms, which are presented in **table 3.1**:

Table 3.1: Acronyms Used in Mobile Routing on the Internet

Acronym	Expansion	Comment
MH	Mobile Host	The host that moves.
CH	Corresponding Host	The host that the mobile is talking to.
HAA	Home Address Agent	The "home" base assigned to the mobile host.
COA	Care-of Agent	The base closest to the mobile host that forwards packets to it.

The basic model for mobile routing, which is similar to the cellular telephone model, is shown in **figure 3.11**. Mobile Hosts (MHs), which are mobile computers with a fixed IP address (much like a cellular phone with a fixed telephone number) communicate with the nearest base station, which is attached to a Care-of Agent (COA).

The care-of agent, which corresponds to a remote MTSO in cellular telephony, receives messages on behalf of the MH. We statically assign each MH to a Home Address Agent (HAA), which corresponds to a local MTSO. We call the machine that the MH is communicating with the corresponding host, or CH. When a corresponding host wants to send a datagram to a mobile host, it puts the mobile host's IP address in the packet destination and hands it to the wide-area network. Using normal network routing, this packet eventually reaches the home address agent. The home address agent is always kept informed of the current care-of agent. It encapsulates the incoming packet with a new header (we shortly see how this is done). It encapsulates the incoming packet with a new header that has its destination set to that of the care-of agent (this is identical to the tunneling used in the MBONE). The care-of agent retrieves the packet and hands it to the base station, which sends it through a wireless link to the mobile host. When the mobile host wants to send a datagram to the corresponding host, it simply puts the corresponding host's IP address in the packet destination, and it is delivered to the corresponding host using normal routing. This solution is nearly identical to the cellular network solution, except that we gain some efficiency in the path from the mobile host to the corresponding host, which does not need to go through the home address agent.

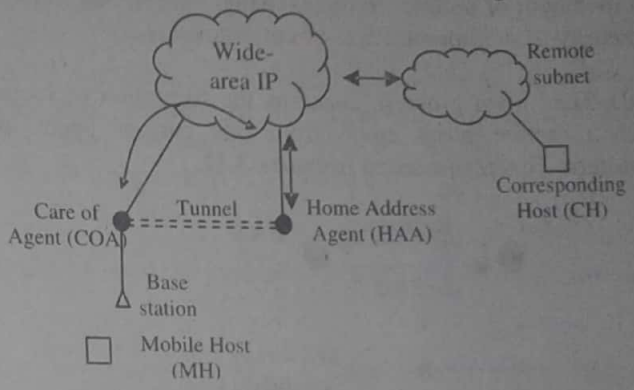


Figure 3.11: Mobile Routing on the Internet

In the **figure 3.11,** packets to a Mobile Host (MH) are always Routed through a Home Address Agent (HAA), which Tunnels Packets for the MH to a Care-of Agent (COA). When the MH Moves, it listens to Beacons to Detect that it has a New COA. It then Updates both the HAA and the Old COA. Packets from the MH use Normal Routing

We still need to solve two problems – first, how does a base station know that it has a mobile host in its area, and second, how does the home address agent find-out the current care-of agent? Beaconing solves the first problem. Each base station periodically transmits a beacon, which is an identification packet transmitted on a well-known frequency. The beacon contains the ID of the base station, and the IP address of the corresponding care-of agent. Each mobile host listens to beacons to detect the best base station to use (depending on the signal strength of the beacon).

It then registers itself with the care-of agent using a registration packet. The care-of agent, in turn, passes on the registration to the home address agent, which then knows which care-of agent to use for the mobile host. If the mobile host moves to a new care-of agent, it knows this from the new base station's beacon, which contains a new care-of agent address. The mobile host sends a registration message to the new care-of agent and "unregisters" itself from the old care-of agent. Thus, the home address agent is aware that the mobile host is now with a new care-of agent. The old care-of agent forwards any packets it receives for the mobile host to the new care-of agent, until the home address agent becomes aware of the change.

Scanned by CamScanner

Security is an important consideration in designing mobile networking protocols. Wireless broadcasts are easy to tap. **Second,** the home address agent has to implicitly trust the care-of agent for carrying packets to the mobile host. A malicious intruder could fake a message from any machine to the home address agent informing it that the intruder's machine was the care-of agent. If the home address agent does not authenticate its messages, the intruder's machine could receive all the packets meant for the mobile host. To prevent this, the mobile host and the home address agent share a common secret, which the home address agent checks before sending packets to the care-of agent. This does not prevent the care-of agent from snooping on the packet, or even prevent a determined cracker from listening to the shared secret and then spoofing it, but adds a modicum of security to the exchange. Improving mobile security is an important area of current research.

A second open problem concerns the formation of loops when mobile hosts move from one care-of agent to another. This is illustrated in **figure 3.12**.

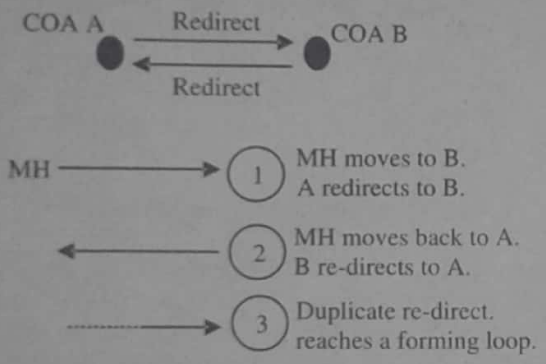


Figure 3.12: Looping in Mobile Routing. If a MH Moves from COA A to COA B and Back, and a Redirection Message is Duplicated, a Loop can Form

Example: Suppose Care-of Agent (COA) forwards packets to the new care-of agent until the home address agent is aware of the change. Suppose a mobile host moves from COA A to COA B (**figure 3.12**). Then, A gets an unregister message from the mobile host asking it to redirect packets to B. If the mobile host moves back to A, the host tells B to redirect packets for it back to A. Now, if a duplicate of an old unregister message reaches A, it forwards messages for the mobile host to B, which forwards it again to A, forming a loop. Since IP does not detect or prevent duplicates, this might well happen. (Routers discard looping IP packets when their time-to-live field eventually drops to zero.)

A third major problem is that packets to the mobile host must always go through the home address agent, even if the packet is generated from a LAN attached to the local care-of agent. **For example,** if the mobile host is running a file transfer with a machine on the care-of agent's LAN, every packet from the corresponding host to the mobile host must still go all the way back to its home address agent, then return to the LAN, which is inefficient. Several schemes for eliminating the extra hops (called the dogleg) from the corresponding host to the home address agent are under active study.

Scanned by CamScanner