Chapter 5

Network Layers

Introduction:

The Network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links). Whereas the data link layer oversees the delivery of the packet between two systems on the same network (links), the network layer ensures that each packet gets from its point of origin to its final destination.

If two systems are connected to the same link, there is usually no need for a network layer. However, if the two systems are attached to different networks (links) with connecting devices between the networks(links), there is often a need for the network layer to accomplish source-to-destination delivery.

(a) 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.

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

Network Layer Design Issues:

The network layer is responsible for routing packets from the source to destination.

- The routing algorithm is the piece of software that decides where a packet goes next (e.g., which output line, or which node on a broadcast channel).
- For connectionless networks, the routing decision is made for each datagram.
- For connection-oriented networks, the decision is made once, at circuit setup time

ROUTING ISSUES:

The routing algorithm must deal with the following issues:

1. Correctness and simplicity: networks are never taken down; individual parts (e.g., links, routers) may fail, but the whole network should not.
2. Stability: if a link or router fails, how much time elapses before the remaining routers recognize the topology change?
3. Fairness and optimality: an inherently intractable problem.
   (a) Definition of optimality usually doesn't consider fairness.
   (b) Do we want to maximize channel usage? Minimize average delay?

Networking Devices:

1. Network Repeater: A repeater connects two segments of your network cable. It sometime regenerates the signals to proper amplitudes and sends them to the other segment. When talking about, ethernet topology, you are probably talking about using a hub as a repeater. Repeaters require a small amount of time to regenerate the signal. This can cause a propagation delay which can effect network communication when there are several repeaters in a row. Many network architecture limit the number of repeater that can be used in a raw. Repeates work only at the physical layer of the OSI network model.

2. Bridge: A bridge read the outmost section of the data packet, to tell where the message is going, it reduces the traffic on other network segment. Since it does not send all packet. Bridge can be programmed to reject packets from particularly network. Bridging occur at data link layer of the OSI model. Which means bridge can not read IP address but only the outermost hardware addres of the packet. In our core bridge can read the ethernet data which gives the hardware address of the destination address, not the IP address. Bridge forward all broadcast message only a special bridge cell translation bridge will allow two network for different architecture to be connected. Bridge do not allow normaly connection of network with different architecture. The hardware address is also called a MAC (Media Access Control) address to determine the network segment:

(a) Transparent Bridging: They build a table address (Bridging table as they receive packets if the address is not in the bridging table the packet is forward to all segment other than the one it come from. This type of bridge used an ethernet network.

(b) Source Route Bridge: The source computer provide path information inside the packet. This is used token ring packet.

(c) Network Router: A router is used to route data packet between two network. It read the information in each packet to tell where it is going if it is designated for an immediate network. It has occurs to it, it will strip the outer packet, readdress the packet to the proper ethernet address, and transmit it on that network. If it is destinated for other network and must to be sent to another router. It will be repackaged the outer data to be received by the next router and send it to the next router. The section of routing explain the theory behind this and how routing table are used to help determine packet destination. Routing occurs at network layer of the OSI model. They connect network with different architecture such as token ring and ethernet. Although they can transfer information from one data format such TCP/IP to another such as IPX/SPX. Router do not send broadcast packets an corrupted packet. If the routing table does not indicate the proper address of a packet, the packet is discarded.

4. Gateway: A gateway can translate information between different network data format or network architecture. It can translate TCP/IP to apple talk so computer supoorting TCP/IP can communicate with apple broadband computer. Most gateway operate at network or session layer but can operate at the application layer but can operate at network or session layer of the OSI model. Gateway will start at lower layer and strip information until it gets to the required level of replacement the information and work its way back toward the hardware layer, of the OSI model. To Confuse issue, when talking about router that is used to interface the another network.

Hubs: A hub is a repeater with more than two parts (i.e. multiport) that is used to interconnect, LAN device. A hub is generally placed in the centre of a network. Hub are sometime called concentrations or wiring concentrators. When they are central component in a star tapology becaue they provide common connection among device since a hub is a layer 1 device. It simply repeats every thing it receives.
A hub, like a repeater is a layer1, device that operates at the bit level and simply regenerates and repeats data. Hub can be passive or active. Passives hubs have number electrical power and therefore can not regenerates signal they simply repeat them. Active hubs are capable of regenerating and repeating data. Because hubs are address independent they can neither filter data traffic nor select the best route to send data. An intelligent hub is a special kind of hub that regenerate and repeats signals and also on onboard processor enabling it to perfrom diagnostic tests and detect when there is a problem with a part.

Switching Technique: Switching in network can be defined as the pathtaken by packets to travel, from source to destinations. It can be connection oriented and connection less path.
There are three different switching technique:
1. Circuit switching
2. Packet switching
3. Message switching

1. Circuit Switching: Circuit switching is a connection oriented service i.e. there is dedicated path from sender to receiver. **eg:** Telephone at the sender and receiver end. There is a need to setup an end to end path before any data can be sent. There is a minimum chance to data lost and minimum error due to dedicated circuit but lot of bandwidth wasted at path cannot be used by other senders during congestion.

An important property of circuit is the need to setup and end to end path before any data can be sent. The elapsed time between the end of dialing and the start of ringing can easily be 10 sec, more for long distance or international calls. During this time interval the telephone system is hunting for a copper path as shown in figure.

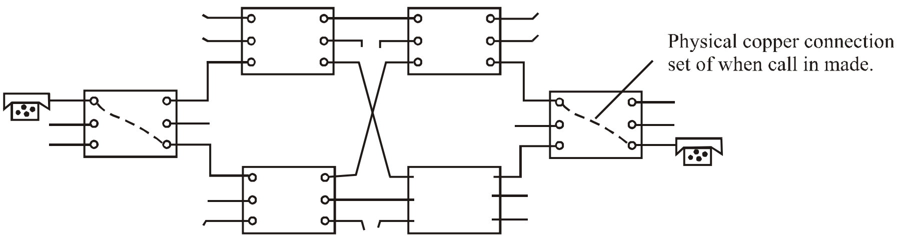
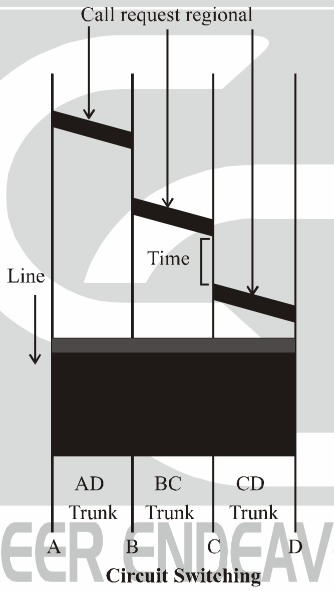


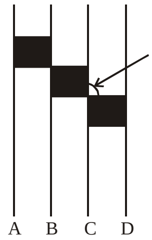
Fig: Circuit Switching in Telephone



Circuit Switching

2. Message Switching: An alternative switching strategy is message swithing when this form of switching is used, no physical copper path established in advance between sender and receiver, when the sender has block of data to be sent. It is stored in first switching office (i.e. router) and then forwarded later one hub at a time. Each block is received in its entirely, inspected for errors and then retransmitted. A network using this technique is called store and forward network.

The first electromechanical telecommunication system used message switching namely for telegram with message switching there is no limit on block size which means router must have disk to buffer long block. It also means that single block may be tie up a router-router line for minutes, rendering message switching useless for interactive traffic. To get around their problems, packet switching was invented.



Message switching

3. Packet switching: Packet switching is a connectionless service i.e. there is no dedicated path between sender and receiver. It places an upper limit on block size. In this type of service bandwidth is freely utilized as unrelated source can used any path. But there are more chance of data lost and error i.e. packet may arrive in wrong order.

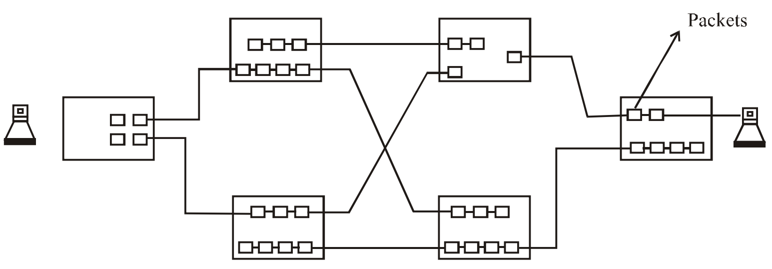
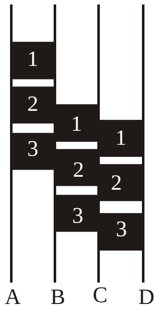


Fig: Packet Switching

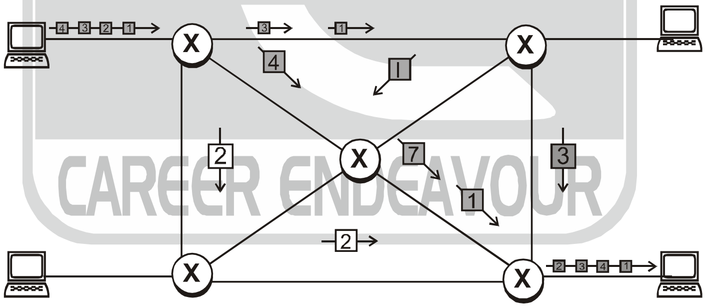


There are two popular approaches to packet switching:
(1) Virtual circuit approach (2) Datagram approach

Virtual Circuit approach: The relationship between all packets belonging to message a session is preserved. A single route is choose between at the belonging of the session when the data are sent all packets of the transmission travel one after another along route. Wide Area Network are the virtual circuit approach to packet switching. The virtual circuit approach needs a call to establish a virtual circuit between the source and destination. A call teardown detects the virtual circuit after a setup routing take place based on virtual circuit identifier.

This approach is used in WAN's frame relay and ATM and is implemented at the data link layer.

Datagram Approach : In the datragram approach each packet is treated independently of all others packet in the approach are refered to as datagram. All the packet belongs to the same message but may go by different paths to reach their destination.



At their destination out of order. In most protocol, it is responsibilty of an upper layer to reorder the datagram before passing them an to the destination port.
The datagram appraoach has some advantage:
(1) It does not need call setup and virtual circuit identifiers.
(2) The routing and delivery of packets are based on the source and destination address include in the packet itself.
(3) The switches or routing each have a routing table that decide on the route based on there two address.

Difference between circuit switching and packet router:
1. Circuit switching reserves the require bandwidth in advance as a result unused bandwidth of allocated circuit is wasted whereas packet switching bandwidth may be utilized by other packet from unrelated source to unrelated destination since there are no dedicated circuit a sudden increase of input traffic may lead to data loss.
2. In circuit switching packets arrive in right order but with packet switching it is possible the packet may or may not arrive in right order.

3. The circuit switching is completely transparent the sender and receiver can use any bit rate format or framing method but in packet switching the carrier determine the basic parameter.
4. Circuit switching have their charge on the distance and time only but in packet switching carriers have their charges an both the number of bytes carried and the connect time.

Comparision between circuit and packet-switching:

Item	**Circuit-Switched**	**Packet-switch**
1. Dedicated copper path	Yes	No
2. Bandwidth	Fixed	Dynamic
3. Potential wasted bandwidth	Yes	No
4. Store-and-forward transmission	No	Yes
5. Each packet follows the same route	Yes	No
6. Call setup	Required	Not need
7. When can congestion occur	At setup time	On every packet
8. Charging	Per minute	Per packet.

CONGESTION:

The network layer also must deal with congestion:
When more packets enter an area than can be processed, delays increase and performance decreases. If the situation continues, the subnet may have no alternative but to discard packets.
If the delay increases, the sender may (incorrectly) retransmit, making a bad situation even worse.

Overall, performance degrades because the network is using (wasting) resources processing packets that eventually get discarded.

INTERNETWORKING:

When we consider internetworking, connecting different network technologies together, there are the same problems, only worse:

- Packets may travel through many different networks
- Each network may have a different frame format
- Some networks may be connectionless, other connection oriented.

Services Provided to the Transport Layer:

Review of Definitions:

Connection-Oriented Service: The subnet, with the help of the network layer, should provide the following operations:

- The sending side of the pair should open a connection with its peer.
- This connection has properties negotiated by the pair.
- Communication is bi-directional and packets are delivered in order.
- Flow control is accomplished in the subnet.

Connectionless Service: The subnet has no state information. It does only send_packet and receive_packet. Error control and flow control are done by the host (network or higher layers.)
Virtual Circuit: Avoids choosing a new route for each packet. A virtual circuit is a state — it remembers how to send a packet from source to destination. This state is held in the subnet.
Datagrams: Each packet sent is routed independently through subnet. Decisions are made with the help of routing table, however this method is more robust.

Routing Algorithms:

Routing algorithms is a part of network layer software which is responsible for deciding output line for incoming packets.

A router has two processes inside it. One of them handles each packet as it arrives, looking up the outgoing line to use for it in the routing tables. The other process is responsible for filling in and updating the routing tables.

There are two types of routing algorithms:

Nonadaptive algorithms - routes never change once initial routes have been selected. Also called static routing.

Adaptive algorithms – Change their routing decisions by using dynamic information as current topology, load, delay, etc.

Either of these algorithms can be applied to either datagrams or virtual circuits.

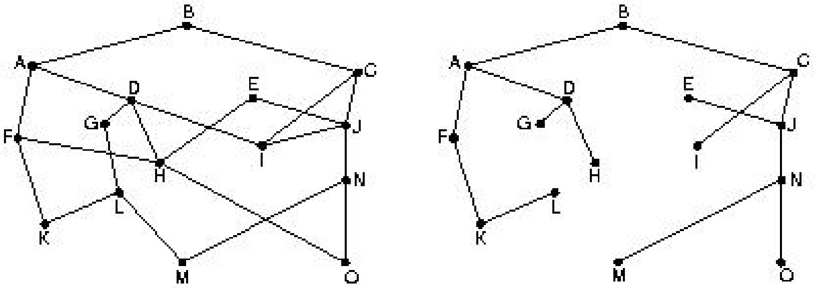
Obviously, adaptive algorithms are more interesting, as nonadaptive algorithms don't even make an attempt to handle failed links.

Optimality Principle:

This simply 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 this same path.

Let 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.

This means we can form a sink tree : the set of optimal path routes from all sources to a given destination form a tree rooted at the destination.



(a) (b)

Routing Algorithm:

Routing is handle delivery of packets. Routing requires a part or a router to have routing table. When a host has packet to send or when a router has received a packet to be forward it takes at this table to find the route to the final destination.

(1) **Next Hop routing Algorithm :** One technique to reduce the content of a routing table is calle d next hop routing. In this technique, the routing table holds only the information that leads to the next hop instead of holding information about the complete route.

PartA

Destination	Route
Part B	R_1R_2, partB

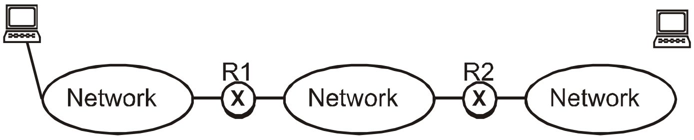
R_1

Destination	Route
Part B	R_2, Part B

R_2

Destination	Route
Part B	PartB

(a) Routing table baud part B on route



Routing table PartA

Destination	Route
Part B	R_1

R_1

Destination	Route
Part B	R_2

R_2

Destination	Route
Part B	-------

(b) Routing table baud on next Hop

(2) **Network Specific Routing :** A record technique to reduce the routing table and simplify the searching process is called network specific routing.

Destination	Next Hop
A	R_1
B	R_1
C	R_1
D	R_1

Destination	Route
Part B	R_2, Part B

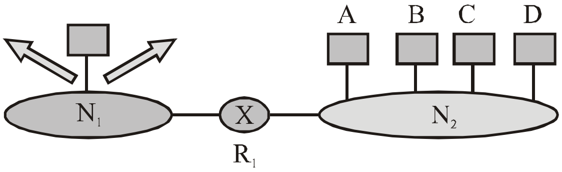
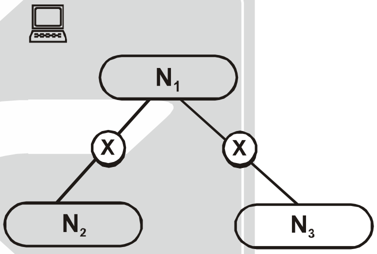


Figure: Network Specific Routing

(3) **Host Specific Routing Protocol Algorithm:** The idea of host specific routing protocol is the inverse of network specific routing.

Destination	Next Hop
PartB	R_3
N_2	R_1
N_3	R3
--------	--------

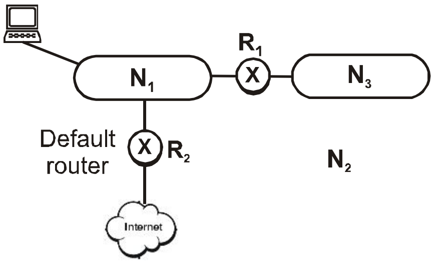


Here, the following is specified for other advantage: Although it is not efficeint to put the host addres in routing table there are occasion in which the administrator that wants to have greater control over routing. for example in fig. if the administrator wants all packets delivered for partB delivered via router R_3 instead of R_1 are single entry in routing table of part A can emplicity define the route.
This algorithm is used for specific purpose such as checking the router security measure.

Default Algorithm : Another technique to specify routing is default routing part A is connected to a network with two router. Router R_1 is used to route the packet to part connected to netwrok N_2, However for the rest of the internet Router R_2 is used. So, instead of lining all the network in the entire internet, Part A can just have one entry called the default.

Static Versus Dynamic Routing: A host of a route keep a routing table, with an entry for each destination to route IP packets. The routing table can be either static or dynamic.

Destination	Net Hop
N_2	R_1
----	----
Default	R_2



Static Routing Table : A static routing table contains information entered manually. The administrator enters the route for each destination in to table. When this type of table is created, it cannot update automatically when there is a change in the internet. The table must be manually altered by the administrators.

A Static routing table can be used in a small internet that does not change very after or in a experimental internet for troubleshooting. It is not good strategy to use a static route table in a big internet such as the internet.

Dynamic Routing Table : A dynamic routing table is updated periodically using one of the dynamic routing protocol such as RIP, OSPF or BGP. Whenever, there is a change in the internet such as shutdown of as route or breaking of a link the dynamic routing protocols update all the table in the routers.

Routing table for the classful addressing : In classful addressing with a without subnetting a routing table needs a minimum of four classes:- mark, destination netwrok address, next hop address and interface is shown in fig.

Mark	Destination Address	Next Hop Address	Interface
L_8	14.0.0.0	118.45.23.8	ml
L_{32}	192.16.7.1	202.45.9.3	m_0
L_{24}	193.14.5.0	84.78.4.12	m_2
L_0	145.11.10.6	145.11.10.6	M_0

Routing table for the classless addressing (CIDR) : The discussion on routing table concentrated an classful addressing, Now we need to consider classes addressing and Classless Inter Domain Routing (CIDR). The shift to classes addressing require changes to the routing table organization and routing algorithm.

Routing table Size : When we are classful addressing, there is only one entry in the routing table for each site outside the organization. The entry define the site even if that site is submitted when a packet carries at the router the router check the corresponding and forwards the packet accordingly.

When we use classful addressing the number of entries in the router table can either decreased or increase. It can decreases the block of address assigned to an organization is larger than the block in classful addressing. for example: Instead of having four entries creates a supernet from four class we can have one entries in classess.

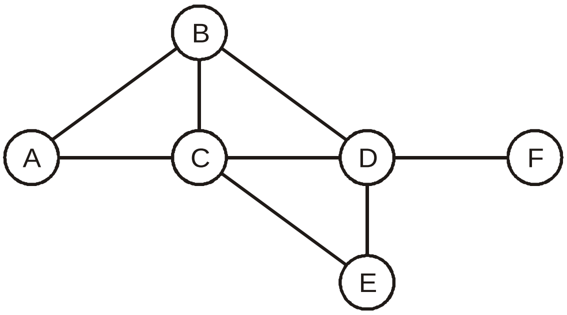
Linked Answer Problem-1 and Problem-2:

Consider a simple graph with unit edge costs. Each node in the graph represents a router. Each node maintains a routing table indicating the next hop router to be used to relay a packet to its destination and the cost of the path to the destination through that router. Initially, the routing table is empty. The routing table is synchronously updated as follows. In each updation interval, three tasks are performed.

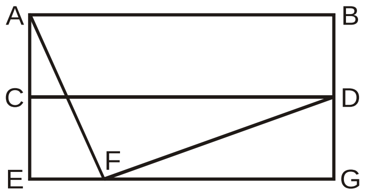
(i) A node determines whether its neighbours in the graph are accessible. If so, it sets the tentative cost to each accessible neighbour as 1. Otherwise, the cost is set to ∞.

(ii) From each accessible neighbour, it gets the costs to relay to other nodes via that neighbour (as the next hop).

(iii) Each node updates its routing table based on the information received in the previous two steps by choosing the minimum cost.



Problem-1: For the network given in the figure below, the routing tables of the four nodes A, E, D and G are shown. Suppose that F has estimated its delay to its neighbors, A E, A and G as 8, 10, 12 and 6 msecs respectively and updates its routing table using distance vector routing technique.



Routing Table of A	
A	0
B	40
C	14
D	17
E	21
F	9
G	24

Routing Table of A	
A	24
B	27
C	7
D	20
E	0
F	11
G	22

Routing Table of A	
A	20
B	8
C	30
D	0
E	14
F	7
G	22

Routing Table of A	
A	21
B	24
C	22
D	19
E	22
F	10
G	0

(a)

A	8
B	20
C	17
D	12
E	10
F	0
G	6

(b)

A	21
B	8
C	7
D	19
E	14
F	0
G	22

(c)

A	8
B	20
C	17
D	12
E	10
F	16
G	6

(d)

A	8
B	8
C	7
D	12
E	10
F	0
G	6

[GATE-2007 : 2 Marks]

Ans. (a)

Soln.

	A	B	C	D	E	F	G
F via A	8	48	22	25	29	0	32
F via B							
F via C							
F via D	32	20	42	12	26	0	34
F via E	34	37	17	30	10	0	32
F via G	27	30	28	25	28	0	6
Min values	≤ 8	≤ 20	≤ 17	≤ 12	≤ 10	0	≤ 6

Based on the given options only (a) matches correct.

Problem-2: For the graph given above, possible routing tables for various nodes after they have stabilized, are shown in the following options. Identify the correct table.

(a)

Table for node A		
A	—	—
B	B	1
C	C	1
D	B	3
E	C	3
F	C	4

(b)

Table for node C		
A	A	1
B	B	1
C	—	—
D	D	1
E	E	1
F	E	3

(c)

Table for node B		
A	A	1
B	—	—
C	C	1
D	D	1
E	C	2
F	D	2

(d)

Table for node D		
A	B	3
B	B	1
C	C	1
D	—	—
E	E	1
F	F	1

[GATE-2005 : 2 Marks]

Ans. (c)

Problem: Continuing from the earlier problem, suppose at some time t, when the costs have stabilized, node A goes down. The cost from node F to node A at time (t + 100) is:

(a) > 100 but finite (b) ∞ (c) 3 (d) > 3 and ≤ 100

[GATE-2005 : 2 Marks]

Ans. (b)

Hierachical Routing : To make the problem of gigantic routing tables we create a same hierarchy in the Internet architecture and create hierarchical routing tables. We mentioned that the Internet today has a same of hierarchy we said that the internet is derived into international and national ISP National ISPs are divided into regional ISPs and regional ISPs are divided into local ISPs if the routing table has a same of hierarchy like the Internet architecture the routing table can decrease in size.

A local ISP can be assigned a single but large black of address with a certain mark the local ISP can be divided this black into smaller blocks of different size and can assigned. There to individual users and organization, both assigened to the local ISP is ABCD/n the ISP can create block of EFGH/m where m may vary for each customer and is greater then n.

Geogrphical Routing :

To decreases the size of the routing even further we need to extend hierarchical routing to include Geographical routing. We must divide the entire address space into a few large block. We assign a a block to North America, a block to Europe, a block to Asia, a block to Africa and so on. The Routers of ISPs outside the europe will have only are entry for packets to europe in their routing tabes. The router of ISP outside of North America in their routing tabls and so on.

Shortest Path Algorithms:

Network routing is a major component at the network layer and is concerned with the problem of determining feasible paths (or routes) from each source to each destination. A router or a packet-switched node performs two main functions: routing and forwarding. In the routing function an algorithm finds an optimal path to each destination and stores the result in a routing table. In the forwarding function a router forwards each packet from an input port to the appropriate output port based on the information stored in the routing table. In this section we present two commonly implemented shortest-path routing algorithms: the Bellman-Ford algorithm and Dijkstra's algorithm. We then present several other routing approaches, including flooding, deflection routing, and source routing.

Most routing algorithms are based on variants of shortest-path algorithms, which try to determine the shortest path for a packet according to some cost criterion. To better understand the purpose of these algorithms, consider a communication network as a graph consisting of a set of nodes (or *vertices*) and a set of links (or *edges*, *arcs*, or *branchesm*), where each node represents a router or a packet switch and each link represents a communication channel between two routers. Figure shows such an example. Associated with each link is a value that represents the *cost* (or *metric*) of using that link. For simplicity, it is assumed that each link is non-directed. If a link is directed, then the cost must be assigned to each direction. If we define the path cost to be he sum of the link costs along the path, then the shortest path between a pair of nodes is the path with the least cost. For example, the shortest path from node 2 to node 6 is 2-4-3-6, and the path cost is 4.

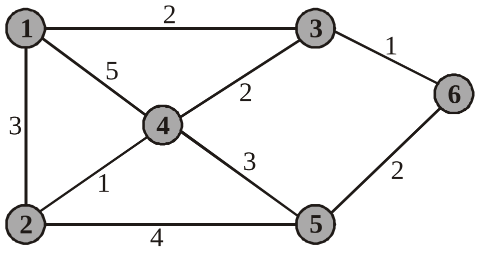


Figure. A sample network with associated link costs

Many metrices can be used to assign a cost to each link, depending on which function is to be optimized. Examples include

1. *Cost* ~ 1 / *capacity*. The cost is inversely proportional to the link capacity. Here one assigns higher costs to lower-capacity links. The objective is to send a packet through a path with the highest capacity. If each link has equal capacity, then the shortest path is the path with the minimum number of hops.

2. *Cost* ~ *packet delay*. The cost is proportional to an average packet delay, which includes queueing delay in the switch buffer and propagation delay in the link. The shortest path represents the fastest path to reach the destination.
3. *Cost* ~ *congestion*. The cost is proportional to some congestion measure, for example, traffic loading. Thus the shortest path tries to avoid congested links.

The Bellman-Ford Algorithm:

The Bellman-Ford algorithm (also called the Ford-Fulkerson algorithm) is based on a principle that is intuitively easy to understand: If a node is in the shortest path between A and B, then the path from the node to A must be the shortest path and the path from the node to B must also be the shortest path. As an example, suppose that we want to find the shortest path from node 2 to node 6 (the destination) in figure. To reach the destination, a packet from node 2 must first go through node 1, node 4, or node 5. Suppose that someone tells us that the shortest path from nodes 1, 4, and 5 to the destination (node 6) are 3, 3, and 2, respectively. If the packet first goes through node 1, *the total distance* (also called total cost) is 3 + 3, which is equal to 6. Through node 4, the total distance is 1 + 3, equal to 4. Through node 5, the total distance is 4 + 2, equal to 6. Thus the shortest path from node 2 to the destination node is achieved if the packet first goes through node 4.

To formalize this idea, let us first fix the destination node. Define D_j to be the current estimate of the minimum cost (or minimum distance) from node j to the destination node and C_{ij} to be the link cost from node i to node j. For example, $C_{12} = C_{21} = 2$, and $C_{45} = 3$ in figure. The link cost from node i to itself is defined to be zero (that is, $C_{ii} = 0$), and the link cost between node i and node k is infinite if node i and node k are not directly connected. For example, $C_{15} = C_{23} = \infty$ in figure. With all these definitions, the minimum cost from node 2 to the destination node (node 6) can be calculated by

$$\begin{aligned} D_2 &= \min\{C_{21} + D_1, C_{24} + D_4, C_{25} + D_5\} \\ &= \min\{3 + 3, 1 + 3, 4 + 2\} \qquad \dots (1) \\ &= 4 \end{aligned}$$

Thus the minimum cost from node 2 to node 6 is equal to 4, and the next node to visit is node 4.

One problem in our calculation of the minimum cost from node 2 to node 6 is that we have assumed that the minimum costs from nodes 1, 4, and 5 to the destination were known. In general, these nodes would not know their minimum costs to the destination without performing similar calculations. So let us apply the same principle to obtain the minimum costs for the other nodes. For example,

$$D_1 = \min\{C_{12} + D_2, C_{13} + D_3, C_{14} + D_4\} \quad ...(2)$$

and

$$D_4 = \min\{C_{41} + D_1, C_{42} + D_2, C_{43} + D_3, C_{45} + D_5\} \quad ...(3)$$

A discerning reader will note immediately that these equations are circular, since D_2 depends on D_1 and D_1 depends on D_2. The magic is that if we keep iterating and updating these equations, the algorithm will eventually converge to the correct result. To see this outcome, assume that initially $D_1 = D_2 = ... = D_5 = \infty$. Observe that at each iteration, $D_1, D_2, ..., D_5$ are non-increasing. Because the minimum distances are bounded below, eventually $D_1, D_2, ..., D_5$ must converge.

Now if we define the destination node, we can summarize the Bellman-Ford algorithm a follows:

1. Initialization

$$D_i = \infty, \ \forall i \neq d \quad ...(4)$$

$$D_d = 0$$

2. Updating: For each $i \neq d$,

$$D_i = \min_j \left\{C_{ij} + D_j\right\}, \forall j \neq i \quad ...(5)$$

Repeat step 2 until no more changes occur in the iteration.

Example - Minimum cost:

Using Figure, apply the Bellman-Ford algorithm to find both the minimum cost from each node to node 6 (the destination) and the next node along the shortest path.

Let us label each node *i* by (n, D_i), where n is the next node along the current shortest path and D_i is the current minimum cost from node *i* to the destination. The next node is found from the value of *j* in equation 5, which gives the minimum cost. If the next node is not defined, we set *n* to –1. Table shows the execution of the Bellman-Ford algorithm at the end of each iteration. The algorithm terminates after the third iteration, since no more changes are observed. The last row records the minimum cost and the next node along the shortest path from each node to node 6.

Iteration	Node 1	Node 2	Node 3	Node 4	Node 5
Initial	(–1, ∞)	(–1, ∞)	(–1, ∞)	(–1, ∞)	(–1, ∞)
1	(–1, ∞)	(–1, ∞)	(6, 1)	(3, 3)	(6, 2)
2	(3, 3)	(4, 4)	(6, 1)	(3, 3)	(6, 2)
3	(3, 3)	(4, 4)	(6, 1)	(3, 3)	(6, 2)

Table - Sample processing of Bellman-Ford algorithm. Each entry for node *j* represents the next node and cost of the current shortest path to destination 6.

Example - Shortest Path Tree:

From the preceding example, draw the shortest path from each node to the destination node. From the last row of Table, we see the next node of node 1 is node 3, the next node of node 2 is node 4, the next node of node 3 is node 6, and so forth. Figure shows the shortest-path tree rooted at node 6.