Contents

Unit 8
The Network Layer

Structure of the Unit

8.1 Unit Outcomes

After the successful completion of this unit, the student will be able to:

- List the benefits of Network Address Translation.
- Outline the features of IPv6.
- Describe the goal of routing in networking.
- Categorize the routing protocols.

8.2 Network Address Translation (NAT)

NAT is another important process followed by a router to map between private and public addresses. Before defining these addresses, let us understand why there is a need for this mapping.

Every device on the Internet must have a unique IP address to communicate with any other device and access its resources. The device can be a home computer, mobile computer, or host connected to an organizational LAN or WAN. The internet service providers (ISPs) of a region distribute the IP addresses in small blocks to an organization or a household. As the existing Internet Protocol version IPv4 uses a 32-bit address field in its IP packet, approximately 4 billion IP addresses can be used. There has been an exponential growth in the number of people using the Internet, and this may lead to the depletion of IP addresses. So, there is a demand for more IP addresses and this issue is being addressed in the next IP protocol version IPv6. IPv6 is discussed in detail in the next section.

IP addresses are categorized into two types: Public IP and Private IP

Public IP address: The Internet Service Providers (ISPs) assign this address to a network router in a region. It is a global and unique IP address and any host on the internet directly access it.

Private IP address: In a private network, this address is assigned by the network router using DHCP. It is unique and local to that network and can be used to connect securely to other devices within that network.

If an organization grows after granting a range of addresses and demands more addresses, the ISP cannot grant, as it could have already used the addresses outside that range. This issue is solved through Network Address Translation (NAT). Here, the devices within an organization use a set of addresses called local private IP addresses and NAT maps these addresses to one unique public or global IP address. The NAT router routes the packets from all the devices through a single connection or global IP and can communicate with the Internet. The private addresses used by one organization can be shared by another organization and they are not visible over the Internet. This non-visibility of the private IP also has another benefit, that is, it aids in securing the private networks form the global internet. Deploying NAT devices also helps a organization in lowering the cost of buying a new IP address for every computer that gets added to the private network of that organization. NAT has load balancing and backup tools which enhances the flexibility and reliability of the network.

The Network address translation (NAT) process is demonstrated in Fig. 8.1
Here a private network is connected to the Internet through a NAT router using one public address (200.24.5.8) and one private address (172.18.3.10). The NAT router hides the private network from being visible on the Internet and only the router can be seen by the Internet.

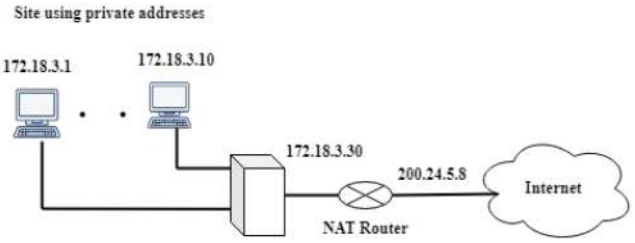


Fig. 8.1: Network Address Translation

8.3 Internet Protocol Version 6 (IPv6)

The devices on the Internet grew with personal computers, smartphones, and various sensor devices as a part of the Internet of Things (IoT). The IPv4 version uses 32-bit addresses and supports approximately 4 billion devices. As the number of Internet users increased, this address space was not enough. So IPv6, the latest version of the IP protocol is used to provide more addresses. It uses a 128-bit address with 2^{128} addresses. Apart from providing a large address space, the IPv6 protocol features efficient handling of packets, improved performance, and increased data security. It also supports the creation of hierarchical routing tables enabling the internet service providers to reduce their sizes.
Example of IPv6 address : 5620:cb:6000:1c82:344c:cc2e:f2fa:5a9b.
It uses a hexadecimal notation with 8, 4- digit hexadecimal numbers separated by colons.

8.3.1 IPv6 Datagram Format

The changes introduced in this version of the IP protocol are seen in the below-given areas:

Addressing, Header contents with available Options, and Packet labeling and Priority.

Addressing:

- It has an expanded addressing capability with 128 bits each allocated to the source and destination addresses.
- It uses anycast address along with unicast and multicast addressing, where a datagram can be delivered to a group of hosts.

Header:

- The header length is 40 bytes and uses options for processing of the datagram faster and also uses a new method of encoding the options in order to process them easily.

Flow labeling and Priority:

- Here, an 8-bit field defines the traffic class. The traffic class can give priority to datagrams of either a specific type of application or datagrams within a flow.

Fig. 8.2 shows the format for the IPv6 datagram header format.

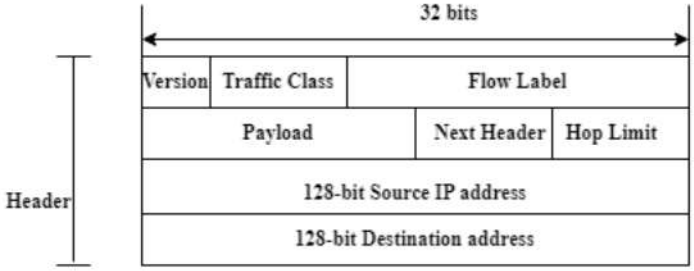


Fig.8.2: IPv6 Header Format

The different fields have the following functions.

Version: This field identifies the version of the protocol and is of 4-bits.

Traffic Class: This field defines the class of the traffic and its size is 8-bits.

Flow Label: This field identifies the flow of the datagram and its size is 20-bits.

Payload Length: This field value specifies the number of bytes present in the datagram after the 40-byte header. Its size is 16-bits.

Next Header: This field specifies two types of values. If an extension header is present, it specifies the type of extension header, else it specifies the transport layer protocol which can be either TCP or UDP. If there is an extension header, it is placed after the fixed header field and before the transport layer header.

Hop Limit: A hop is the process of the datagram movement between two routers, so the hop count gives the number of routers through which a datagram is forwarded from its source until it reaches its destination. The hop limit is an 8-bit field containing an integer value that gets decremented by 1 at the router which forwards the Datagram. If the datagram is not delivered to its destinatio and hop count reaches zero, then the datagram gets discarded.

Source and Destination Addresses: These are 128-bit addresses specifying the source and destination addresses with an address space identifying approximately 3.4×10^{38} addresses.

Data: This specifies the data or payload portion of the datagram which has to be delivered to the destination host.

Extension Headers:

Some of the fields of IPv4 such as Fragmentation/Reassembly, Header Checksum, and Options have been removed in IPv6. These fields are added in the extension headers if needed. Most of the extension headers are not processed at the intermediate routers but only at the destination host. This reduces the processing time of the datagrams at all the routers and improves the performance.

The datagrams are not fragmented at the intermediate routers but these operations are performed at the source and destination. If a router receives a datagram of large size and is unable to forward it, it is dropped.

The Checksum field is not added in IPv6 headers, as the transport layer calculates the checksum at the source and uses it by the receiver to verify for errors in transmission or packet corruption. If added in the IP packet, the checksum has to be recomputed at every router. This would increase the packet processing time.

The other extension headers which are currently defined in the IPv6 are: Routing, Authentication, Encapsulation, Hop-by-hop option and destination options.

8.3.2 Transitioning from IPv4 to IPv6

The IPv6-based systems can be made to be backward compatible to handle IPv4 packets, but the reverse is not easy. There have been many options to make this transition possible. As the Internet has grown tremendously, upgrading millions of machines, network administrators and users would be a very great task.

The first option thought of was the declaration of a flag day when all the machines on the Internet were to shut off on a specific day and time and upgrade themselves. This option was not feasible 25 years ago even when the Internet was very small for upgrading NCP to TCP at the transport layer.

The second option was to use a dual-stack approach with nodes capable of both IPv6 features and a full fledged IPv4 implementation. So such a node can send and receive both IPv4 and IPv6 datagrams. These nodes should have both IPv4 and IPv6 addresses for their identification on the Internet and the nodes should be identified for sending either IPv4 or IPv6 datagrams. Using a Domain Name System (DNS) this problem can be solved, but if the machine issuing a DNS request can identify only IPv4 datagrams, then it can only return an IPv4 address. Also, with an appropriate address mapping done, an IPv6 datagram can be copied in the data field of an IPv4 datagram and sent to an IPv4 capable node. However, the reverse is not possible.

The above situation can be demonstrated with the help of an example shown in Fig. 8.3. Here, a node A which is an IPv6 node wants to send a datagram to node F which is also an IPv6 node but the datagram has to pass through two IPv4 nodes C and D. Here, transmission from node A to B and E to F is

straightforward, but transmissions from B to C and D to E need two conversions. In the first conversion, the contents of the data field of the IPv6 datagram are copied into the data field of an IPv4 datagram at node B. After performing an appropriate address mapping, the datagram is sent to node C. This poses no issues. But, while converting an IPv6 datagram to IPv4 at node D, some fields in IPv6 cannot map with IPv4 as some IPv6-specific fields in the IPv6 datagram have no appropriate counterpart in IPv4. This issue can be overcome using Tunneling.

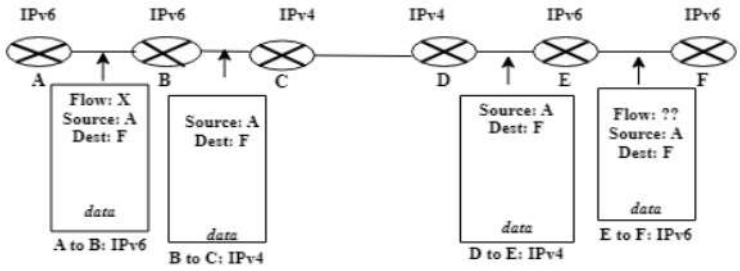


Fig. 8.3: A dual - stack approach

The third possible approach is a technique called Tunneling. The working can be explained through Fig. 8.4 which uses the same example as in Fig. 8.3. The IPv4 routers between C and D are called intervening routers or a tunnel. At node B, the complete IPv6 datagram is copied inside the payload field of an IPv4 datagram and is addressed to node E, as this node is on the receiving side of the tunnel. The IPv4 datagram is then sent to the first node in the tunnel which is node C. Inside the tunnel, the datagram is routed as an IPv4 datagram in the IPv4 network. At the end of the tunnel, the IPv6 datagram is extracted and is then routed as an IPv6 datagram.

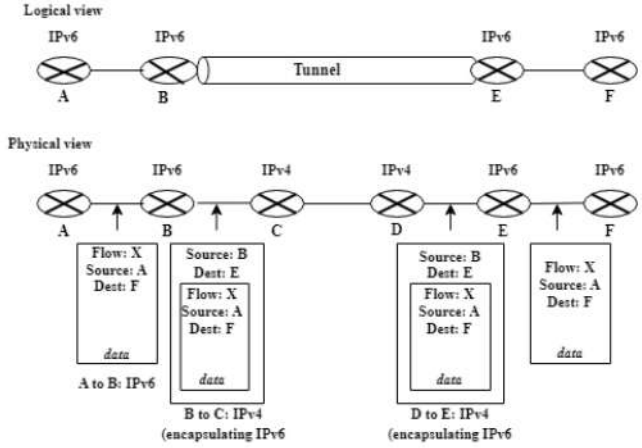


Fig. 8.4: Tunneling

8.4 Routing Algorithms

Out of the two important functions of the network layer: Forwarding and Routing, we have already discussed the function of forwarding a packet, where the router must move the incoming packet to an appropriate output link. In this section, we will study how a network layer determines the best path to be taken by a packet to flow from its source host to its destination, through the network of routers. The

algorithms to find these paths or routes are called routing algorithms and the process is called **Routing**.

If a packet is sent to only one destination, it is called Unicast routing and if the packet is destined to multiple hosts, it is called Multicast routing.

Routing algorithms run in the network routers and are used to compute and exchange the information which helps in configuring the forwarding tables. The values in the tables are inserted by these algorithms.

Routing Tables:

A "forwarding table" is a generic term and includes Layer 2 forwarding, and other forwarding technologies like Multi-protocol Label Switching (MPLS) forwarding, and Policy-based routing (PBR) forwarding. It is not always strict IP routing. But a "routing table" represents a Layer-3 forwarding table based on IP. It is a subset of a forwarding table. That is, all routing tables are a form of forwarding tables.

A routing table has three important information fields: network identifier, metric, and next-hop address which are compulsory, and other information based on the implementation and applications.

Network identifier: Specifies the destination subnet and mask

Metric: Specifies the routing cost of the path which the packet is going to follow.

Next-hop address: It is the address of the next node that the packet should use to reach its destination. With the help of the packet's destination IP address, the router determines the "next hop" IP address. A switch uses the "next hop" IP address in the forwarding tables to find an interface to which it should deliver the packet to that next hop. In this process, it also determines the MAC address of the node which receives the packet.

Eg: Multipoint interfaces like Ethernet or Wi-Fi.

The routing algorithms may run at a central site or may be decentralized and can run on all the routers. Irrespective of whether the network layer provides a connectionless or connection-oriented service, the routing process is performed with the help of the forwarding tables in the routers.

8.4.1 Categories of Routing Algorithms

There are three ways in which the routing algorithms are categorized:

1. Based on the location where they run.
2. Based on how they change the routing paths over time.
3. Based on whether they are sensitive to load changes.

1. Based on the location where the routing algorithms run, they are of two types: Global (Centralized) and Decentralized.
2. Based on how they change the routing paths over time, they are of two types: Static and Dynamic.
3. Based on whether they are sensitive to load changes, they are of two types: Load-sensitive or Load-insensitive

8.4.1.1 Global and Decentralized

Global routing algorithms use the global knowledge of the network to compute the least-cost path between a source host and a destination host. They use the knowledge about how all the nodes and links are connected in order to compute the paths. The computation can run at one centralized location or may be replicated at many sites. Such algorithms with global state information are called Link-state (LS) algorithms.

Decentralized routing algorithms, calculate the least-cost paths in an iterative and distributed manner. Here, initially none of nodes has a complete information about all link costs. But computation starts with each node having knowledge about the cost of its directly attached links. A node then gradually, computes the least-cost path to one destination or a set of destinations by calculating and exchanging information with its neighbors through an iterative process. One of the decentralized algorithms we will be discussing is the Distance-vector (DV) algorithm.

NOTE: Both LS and DV algorithms are Unicast routing algorithms.

8.4.1.2 Static and Dynamic

Static Algorithms are also called non-adaptive algorithms. The routes calculated by a static algorithm change very slowly over time because they are manually calculated. When a network gets booted, the routing information gets loaded on the routers. These algorithms are very simple and work well with stable loads and reliable networks, but they do not perform well when the traffic volume increases or network topologies change.

There are two types of static algorithms: Flooding and Random Walks

In the case of Flooding, an incoming packet is sent to all the outgoing links except the link from where the packet arrived. This can result in duplicate packets being received by some nodes.

In the Random Walks algorithm, an incoming packet is sent to one of its neighbors randomly, so it makes use of alternative routes efficiently.

Dynamic routing algorithms are also known as adaptive routing algorithms. These change the routing paths whenever there are changes in traffic loads or the network topology. The routing information is provided to a router by its adjacent routers or all the routers. Through these algorithms, the hop count and transit time of a packet are optimized. These algorithms run either periodically or when there is a change in the topology or link costs. The Global and Decentralized routing algorithms are adaptive.

8.4.1.3 Load-sensitive or Load-insensitive

In a Load-sensitive algorithm, link costs change whenever there is a change in the congestion levels in a particular link. If a congested link cost is high, a path is chosen to avoid that path.

Eg: ARPnet.

As these algorithms were load-sensitive, they faced problems while calculating the routes.

So, today's Internet routing algorithms such as Routing Information Protocol (RIP), Border Gateway

Protocol (BGP), and Open Shortest Path First (OSPF) are used which are load-insensitive and do not reflect the current levels of congestion.

8.4.2 Goal of a routing algorithm

Most of the time, a host is connected directly to a router. This router is called the default router or the first-hop router. At the sending end, it is called the source router, and at the receiving end, it is called the destination router.

Fig.8.5 shows a small network of routers with six routers, R1 to R6. Here, for the source host, R1 is the source router and R4 is the destination router.

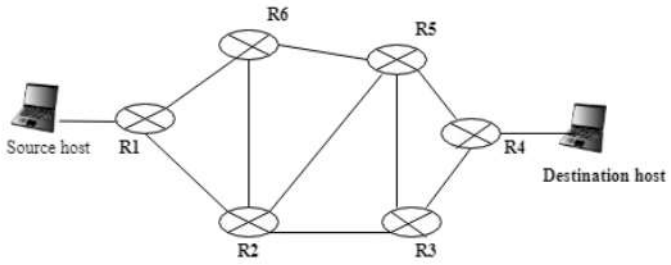


Fig. 8.5: A network of routers

The main goal of a routing algorithm is to find the best path from the source router to the destination router. The path can be called as best **if the cost of the path is the least.**

NOTE: Cost can be associated with link speed, hop count, or time delay.

8.4.3 Internet as a Graph

To understand how a routing algorithm is formulated, a graph model is applied to a computer network as shown in Fig.8.6.

We know that a graph represented by G = (N, E) has a set of nodes N with a set of edges E. An edge is a pair of nodes from N. In the routing context in computer networks, the nodes in the graph represent the routers and the edges represent the physical links connecting two routers. As each edge is associated with a cost, the Internet is modeled as a weighted graph.

In the figure, the value shown on each edge represents a cost. The cost may represent the link speed, the link length, or a monetary cost associated with it. For an edge (x,y) belonging to E, c(x,y) represents the cost between the nodes x and y and for an edge (x,y) not belonging to E, its cost is ∞.

For simplification, we assume a cost is just a number, and the graph is undirected so c(x,y) = c(y,x). Also, node x is the neighbor of node y.

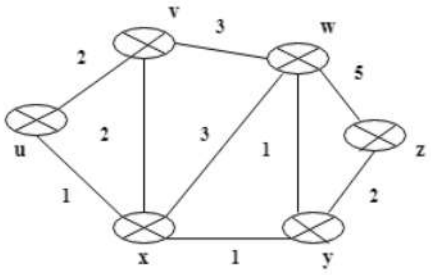


Fig. 8.6: Abstract graph model of a computer network

If the sequence of nodes is w_1, w_2, w_3, ….. w_n, and the costs of the path are $c(w_1, w_2)$, $c(w_2, w_3)$, …… $c(w_{n-1},w_n)$ for node pairs (w_1, w_2), (w_2, w_3), …. (w_{n-1},w_n) respectively, then the cost of a path from source w_1 to destination w_n is given by, $c(w_1, w_2) + c(w_2, w_3) + \ldots\ldots + c(w_{n-1},w_n)$.

8.4.4 Least Cost Routing

There will be many paths between the source and the destination. Out of all the paths, there will be one or more paths having the least cost. Each router has to find the least-cost route between itself and all the other routers in the Least Cost Routing algorithm.

In Fig. 8.6, the least-cost path between nodes u and w is (u,x,y,w) = 3.

NOTE: If all the edges in a graph have the same cost, then the path with the least number of links is the least-cost path or the shortest path.

8.4.4.1 Least-Cost Trees

In a network with N routers, there will be (N-1) least-cost paths between one router to all other routers. So, if there will be N(N-1) least cost paths for the whole network. A least-cost tree is a tree with the source router as the root with all other nodes as its branches, and each path between the source and every other node is the shortest. So there will be N least-cost trees for a network of N nodes.

Let us understand this concept with a graph with three nodes A, B, C, and the least-cost trees for all the nodes in Fig. 8.7

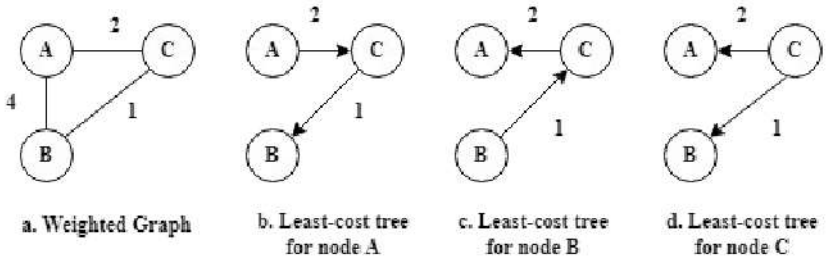


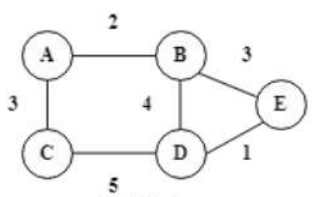
Fig. 8.7: A Weighted Graph and its least-cost-trees

8.4.5 The Link-State Routing Algorithm

This routing algorithm incorporates least-cost trees and forwarding tables to find the best routes from each node to every other node. Here, the state of the link is defined by the cost associated with an edge. Lower-cost links are preferred over higher ones and a link with an infinite cost is either broken or does not exist.

8.4.5.1 Link-State Database (LSDB)

This is a huge collection of the state of all the links in a network. The Internet has one large LSDB. For a node to create a least-cost tree, it needs to have a copy of this LSDB. Fig.8.8 shows a weighted graph of five nodes and their LSDB.



a. Weighted graph

	A	B	C	D	E
A	0	2	3	∞	∞
B	2	0	∞	4	3
C	3	∞	0	5	∞
D	∞	4	5	0	1
E	∞	3	∞	1	0

b. Link state database

Fig.8.8: Example of a link-state database

8.4.5.2 Creation of an LSDB with the help of each node

This is done by a process called **flooding.** Here every node sends greeting messages to its neighbors and collects two pieces of information: the node's identity and the cost of the link. These two pieces of information are put up in a packet called a link-state (LS) packet (LSP). The LSP is sent out on every link interface. This LSP is compared with the existing copy of the LSP. If the received LSP is older than the already existing one, then it is discarded otherwise, it keeps the new LSP. The node then sends a copy of the new LSP to all the interfaces excluding the one from where it received the LSP. So after receiving all new LSPs, each node helps in creating the LSDB as shown in Fig.8.9.

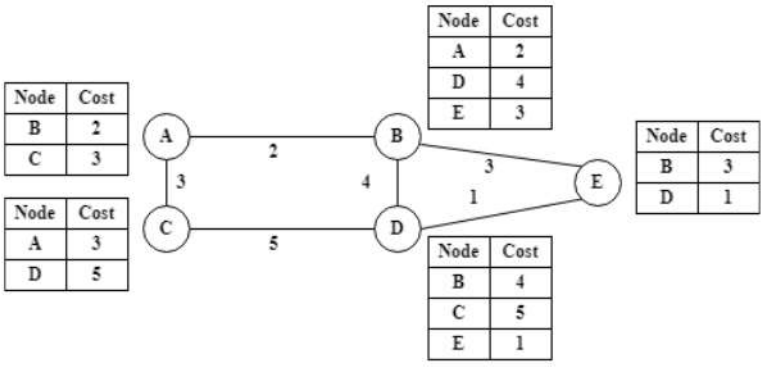


Fig.8.9: LSPs created and sent out by each node to build LSDB

8.4.5.3 Formation of Least-Cost (or Shortest Path) Trees

To create the least-cost tree for itself, here every node in the graph uses the shared LSDB and Dijkstra's algorithm. Dijkstra's algorithm runs iteratively on every node and is used to find the shortest paths from the source node to all other nodes in a network.

It uses the following steps:

1. The node creates a tree with a single node with itself as the root.
2. Based on the information in the LSDB, the total cost of each node is set.
3. The node then selects a node closest to itself and adds it to the tree.
4. The cost of all the other nodes which are not in the tree is updated.
5. Steps 3 and 4 are repeated until all the nodes are added to the tree.

Dijkstra's algorithm: Finds the shortest path tree

Let N: Set of nodes for which the shortest path is found.

Let Di be the current minimum (distance) cost from the source node s to node i.

1. Initialization (start with the source node s)
   N = {s}
   Ds = 0 (as s is distance zero from itself)
   Dj= Csj, for all j ≠ s (distances of directly connected neighbors)
2. Find the closest node: find node i not belonging to set N,
   Dj = min Dj
   Add i to N
   If N contains all nodes, stop
3. Update minimum cost after node i is added to N, for each j not belonging to set N,
   Dj = min{ Dj + Cij }
4. Go to step 2

Let us understand the algorithm with an example given below.

Example 1: Find the shortest path tree at node A for the network graph given in Fig. 8.10

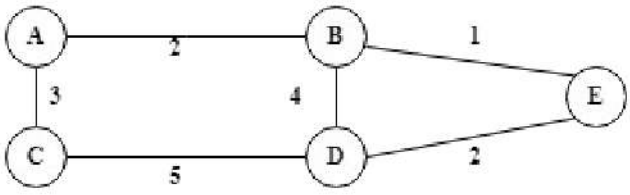


Fig.8.10: Example network for finding shortest path tree

Solution 1: Table 8.1 shows the iterative steps in finding the shortest path from A to all the other nodes.

Table 8.1: Iterations in finding the shortest path

Iteration	N	D_B	D_C	D_D	D_E
Initial	{ A }	2	3	**8**	**8**
1	{ A, B }	2	3	6	3
2	{A, B, C}	2	3	6	3
3	{A, B, C, E}	2	3	5	3
4	{A, B, C, D, E}	2	3	5	3

The shortest path tree is given in Fig.8.11.

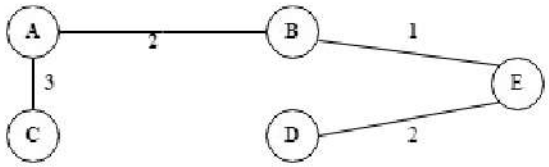


Fig.8.11: Shortest path tree using Dijkstra's Algorithm

8.4.6 The Distance Vector Routing Algorithm

This algorithm is asynchronous, iterative, and distributed. Asynchronous because all the nodes do not operate in synchronization with each other. It is distributed because every node receives information from its directly attached nodes, makes computations, and then distributes the results to its neighbors. It is iterative because the procedure repeats until no new information is exchanged between the neighbors.

This algorithm uses an important relationship between the costs of the least-cost paths given by Bellman-Ford.

The Bellman-Ford algorithm is used to find the minimum cost of each node to the destination node and the next node along the shortest path.

Node A can determine its shortest path to node Z if all neighbors of node A know the shortest path to node Z. Node A calculates the cost/distance to node Z from each of its neighbors and then selects the minimum of all the values.

The Bellman- Ford Algorithm:

Let i represent all nodes other than the destination node and d represent the destination node.

Cij = link cost from node i to node j.

Each node maintains an entry (n, Di), where n is the next node along the shortest path and Di is the current minimum distance from node i to node d.

Algorithm:

1. Initialization

   $Di = \infty$, for all $i \neq d$ (destination node to all nodes)

   Dd = 0 (destination node to itself)

2. Updating (finding min. distance to destination through neighbors)

for each $i \neq d$, $Di = \min \{ Cij + Dj \}$, for all $j \neq i$

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

Let us understand the algorithm with an example given below.

Example 2: Find the minimum cost of all the nodes to the destination node E for the graph given in Fig. 8.10.

Solution 2: Table 8.2 shows the iterative steps in minimum cost from all nodes to node E.

As initially the next node n along the shortest path is not known, it is taken as -1.

Table 8.2: Iterations in finding the minimum cost

Iteration	Node A	Node B	Node C	Node D
Initial	(-1,∞)	(-1, ∞)	(-1,∞)	(-1,∞)
1	(-1,∞)	(E,1)	(-1, ∞)	(E,2)
2	(B,3)	(E,1)	(D,7)	(E,2)
3	(B,3)	(E,1)	(A,6)	(E,2)

8.5 Unicast Routing Protocols

Routing protocols are a set of rules and algorithms running on all the routers on the Internet. They are used to communicate with each other to exchange routing information. They are divided into two types: Interior Gateway Protocols (IGP) and Exterior Gateway Protocols (EGP).

Interior Gateway Protocols are used to route packets within a single autonomous system (AS) or domain, whereas an Exterior Gateway Protocol is used to route packets between different autonomous systems or domains. An autonomous system is a group of routers under a common administrative control.

In the following subsections two Interior Gateway Protocols, Routing Information Protocol (RIP) and Open Shortest Path First (OSPF), and one Exterior Gateway Protocol, Border Gateway Protocol (BGP) are discussed. These are commonly used on the Internet today.

8.5.1 Routing Information Protocol (RIP)

This protocol is based on the Distance Vector Routing algorithm and has two versions one started by Xerox Network System (XNS) and the other Unix version by Berkeley Software Distribution (BSD).

The routing metric used by this protocol is the hop count which uses a maximum hop count of 15 to prevent routing loops, but this in turn reduces the size of the network that the protocol can support. This protocol assumes that the destination is unreachable if the hop count is 16 or more.

In RIP version 1, an RIP-implemented router broadcasts a request message for routing information at startup and every 30 seconds thereafter through the RIP-enabled interface. The neighboring routers

respond to this request by sending a segment containing the routing table. The requesting router will update its routing table with new information such as a hop count to a reachable network. If there are two routes found to the same network with the same hop count, the router tries to compare the cost of each path to route the packets. The router also listens to incoming requests and sends their routing tables for the neighboring routers to update. RIP uses UDP as its transport protocol at port number 520. RIP version 1 uses classful routing and does not carry subnet information. This makes the use of different-sized subnets inside the same network class difficult. Also, RIP is prone to different attacks as it does not support router authentication. The RIP version 2 was developed to overcome the above limitations. It also uses route tags to differentiate between the routes learned from RIP and other protocols.

8.5.1.1 Hop Count

In this protocol, two modifications are done to the Distance Vector Routing Algorithm, the first one is to route packets between different networks in addition to routing between two hosts in the same network, and the second one is to simplify the link cost implementation and make it independent of the bandwidth, delay, etc. which are the performance parameters of the links and routers.

Fig. 8.12 demonstrates the concept of hop count advertised by two routers R1, and R2 from a source host to a destination host for a configuration consisting of three networks N1, N2, and N3 connected by these two routers.

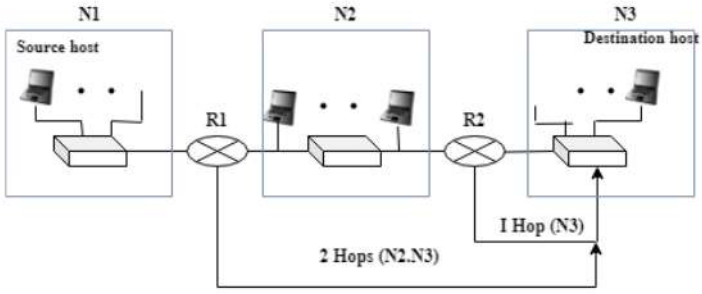


Fig. 8.12: Hop Count in RIP

8.5.1.2 Forwarding Tables

Here the forwarding table is three columned. They contain the destination network address, the next router address, and the hop cost. The second field which is the next router gives the information of the complete least cost tree. The third column is not used for the forwarding process but it is used to update the forwarding table whenever there is a change in the route.

Table. 8.3 shows the forwarding tables for the example network configuration shown in Fig. 8.12.

Table 8.3: Forwarding tables

Forwarding table for R1

Destination Network	Next router	Cost in hops
N1	------	1
N2	------	1
N3	R2	2

Forwarding table for R2

Destination Network	Next router	Cost in hops
N1	R1	2
N2	-------	1
N3	-------	1

8.5.1.3 RIP Implementation

This protocol uses the services of UDP at the transport layer on port number 520. In the BSD version, RIP runs as a daemon process in the background, and as a routing protocol helps the IP to route its datagrams through the autonomous systems. This protocol uses messages which are encapsulated inside the UDP user datagrams which are further encapsulated inside the IP datagrams. So, RIP creates forwarding tables for the IP at the network layer but runs at the network layer.

8.5.1.4 RIP Messages for Version 2

Messages need to be exchanged between the Client and Server RIP processes for RIP working. Fig. 8.13 shows the RIP message format. The format shows the contents of one row of the forwarding table only and this is represented by the name "entry". So, based on the row entries in the forwarding table, the entries will be repeated.

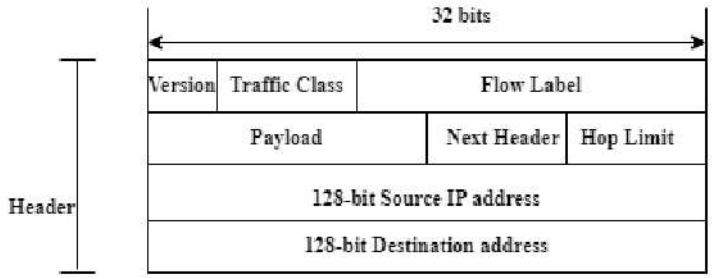


Fig.8.13: IPv6 Header Format

RIP exchanges two types of messages: request and response.

Request message:

- Is sent by a router that has some time-out entries or when it has just booted up.
- Is sent to enquire about one or more entries in the forwarding table.

Response message:

- A solicited message can be sent in response to a request message to give information about the destination in the request message.
- An unsolicited message is sent whenever there is a change in the forwarding table and also periodically every 30 seconds.

8.5.1.5 RIP Algorithm

This algorithm has some modifications done to the distance-vector routing algorithm for a router to update its forwarding table as given below:

- In a response message, a router sends the complete forwarding table.
- The received router adds one hop to each of the costs in its table and the address of the sending router to the next router field. The new routes are called as received routes and the unmodified ones as old routes.

The new routes will be followed only when,

- There is no entry for a received route in the old forwarding table.
- The cost of a new route is less than the old one.
- The cost of a new route is more than the old one, but the next router is the same in both routes.

8.5.2 Open Shortest Path First (OSPF)

This protocol is a Link State Routing Protocol for IP networks and operates within a single autonomous system controlled by a single administrative body with well-defined routing policies. This protocol constructs a topology map of the network after gathering link state information from all the available routers. This topology map is presented in the form of a routing table to the Internet layer for routing packets to their destination. OSPF supports IPv4 and IPv6 networks and CIDR addressing model.

8.5.2.1 Metric

The routing metrics can be the cost factor associated with every link interface. The cost factor is based on the data throughput of a link, round-trip time, reliability, or sometimes hop count, and is expressed as an integer.

Fig. 8.14 shows the Metric in OSPF for the same network configuration discussed in RIP.

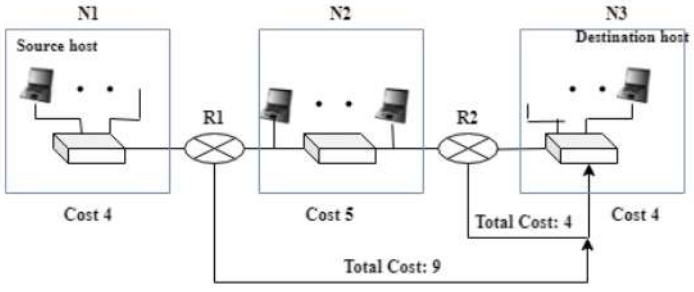


Fig. 8.14: Metric in OSPF

8.5.2.2 Forwarding Tables

As it is a link-state routing protocol, it uses Dijkstra's algorithm to find the shortest path tree between the destination and itself and then creates forwarding tables based on the tree. These tables are similar to the RIP forwarding tables except for the cost-value difference. It can detect link failures and converge into a loop-free routing structure in a very short time.

Table 8.4 shows the forwarding tables in OSPF for the configuration demonstrated in Fig.8.14

Table 8.4: Forwarding tables in OSPF

Forwarding table for R1

Destination Network	Next router	Cost
N1	------	4
N2	------	5
N3	R2	9

Forwarding table for R2

Destination Network	Next router	Cost
N1	R1	9
N2	-------	5
N3	-------	4

8.5.2.3 Areas

OSPFs are designed for larger ASs. For creating the global LSDB, the shortest-path tree is formed and this requires flooding by all the routers with their LSPs. This creates a huge amount of traffic in larger ASs. To reduce this traffic, larger ASs are divided into smaller areas with independent domains for routing. So there are two levels of hierarchy in routing: autonomous system and area.

As all the routers in an area need to know the link state in its area and other areas, one area is designated as the backbone and this area connects all the areas together. The routers in this area pass the information collected by each area to all the other areas.

8.5.2.4 Link-State Advertisement

For forming the LSDBs, OSPF uses a link-state algorithm where the routers advertise the state of each link to all its neighbors. In the real world, the routers advertise the existence of the nodes, the state of the types of links connecting the nodes, and the types of costs associated with each link. For this purpose, there are five link-state advertisements used: router-link, network-link, summary link to network, summary link to AS border router, and external link.

8.5.2.5 OSPF Implementation

There are three versions of OSPF. Version 2 supports IPv4 and Version 4 supports IPv6 networks. This

protocol is implemented at the network layer in the form of a program. It does not use UDP or TCP at the transport layer but uses the services of IP for its propagation and incorporates its own error detection and correction functions. It encapsulates its data in IP packets with protocol number 89.

8.5.2.6 OSPF Messages

This protocol uses five different types of messages used by a router:

1. Hello message: Used for introducing itself to its neighbors and giving known information.
2. Database description message: Sent in response to the hello message.
3. Link-state request message: Sent to get information on the state of a link.
4. Link-state update message: Used for building the LSDB.
5. Link-state acknowledgment message: Used to acknowledge the receipt of link-state information.

8.5.2.7 OSPF Algorithm

This algorithm has some modifications done to the link-state routing algorithm as given below:

- The algorithm has to create a corresponding routing algorithm for every shortest path tree created.
- The algorithm has to implement the different message-handling functionality in it.

8.5.2.8 Performance of OSPF

- As discussed earlier, the bigger the area, the heavier will be the traffic created due to flooding of the link state information. This leads to increased bandwidth usage.
- Convergence can be faster due to each router creating its shortest-path tree and forwarding table. However, running of the Dijkstra's algorithm at each router may consume time.
- OSPF is more robust compared to RIP because once the router updates its LSDB, it can route datagrams independently, and as in RIP, the failure of one router does not seriously effect the other routers.

8.5.6 Border gateway Protocol (BGP)

Today, this is the only interdomain routing protocol used on the Internet and its basis is the path-vector algorithm. In this algorithm, the goal for routing is not based on least cost but on the policies imposed on the route by the source. The path-vector algorithm controls the path for routing and is used on the Internet to route packets between ISPs. This algorithm uses the best spanning tree to determine the path from a source to all destinations.

To understand this protocol, we use Fig. 8.15 with a sample Internet with three autonomous systems AS1, AS2 and AS3. These ASs use either of the intradomain routing protocols: RIP or OSPF. The routers in these ASs have the knowledge about reaching a network in its own AS but not in another AS. In order to route packets between two ASs, all the edge routers or border routers run one version of a BGP4 protocol called external BGP (eBGP) and another version of the BGP4 called internal BGP (iBGP) are installed on all the other routers inside an AS. So, the border routers run three protocols: an

intradomain, eBGP and iBGP protocol. Whereas, the other routers run an intradomain and iBGP protocol.

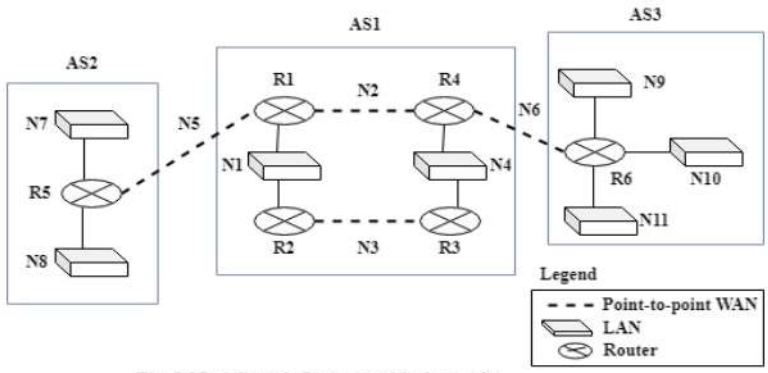


Fig. 8.15: A Sample Internet with three ASs

8.5.6.1 Operation of External BGP (eBGP)

BGP protocol is a type of point-to-point protocol. After establishing a TCP connection between two routers on port no. 179, communication between these routers takes place by exchanging messages through a pair of client and server processes. The BGP routers running the BGP processes are called BGP peers or BGP speakers.

In the eBGP version, the physical connection between the border routers forms a pair of BGP peers in two different ASs. In the sample Internet shown in Fig. 8.15, R1-R5, R4-R6 are the two pairs of eBGP peers. The connection between these pairs is established through the physical WANs N5 and N6. The logical connection over which the exchange of information take place between the ASs is called a session. So there are two eBGP sessions involved as shown in Fig. 8.16.

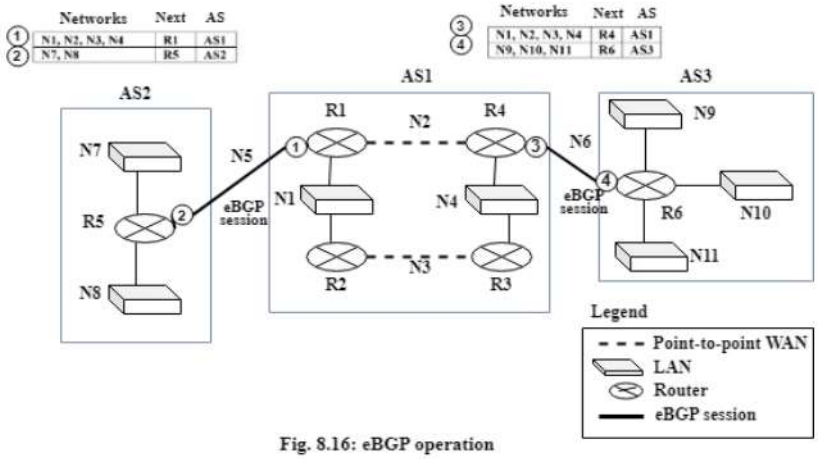


Fig. 8.16: eBGP operation

Messages 1, 2, 3 and 4 are sent by the routers R1, R5, R4 and R6. Message 1 tells the router R5 that networks N1, N2, N3 and N4 are reachable through R1. This information is extracted by R1 through the corresponding intradomain forwarding table and R5 adds this information to its forwarding table and uses it to forward the incoming packets destined to N1, N2, N3 and N4.

The eBGP has the limitation that some border routers do not have information to route packets to non-neighbor ASs and also none of the non-border routers have the information to route packets for any networks in other ASs. These limitations are removed in the second variation of the BGP protocol, iBGP.

8.5.6.2 Operation of Internal BGP (iBGP)

This protocol uses the services of TCP on port 179 to create a session between a pair of routers within an autonomous system (AS). But there are two important points regarding the iBGP sessions given below:

- There is no iBGP session created in an AS with only one router.
- There will be [n x (n-1)/2] iBGP sessions in an AS with n number of routers.

Here, each router advertises its own reachability to the peer in the session and does not flood the network with the information it receives from another peer in another session. Fig. 8.16 shows the combination of eBGP and iBGP for the sample internet. As a session is made on a logical TCP connection spanning many physical networks determined by the route given by the intradomain routing protocol, the physical network inside the ASs is not shown in the figure.

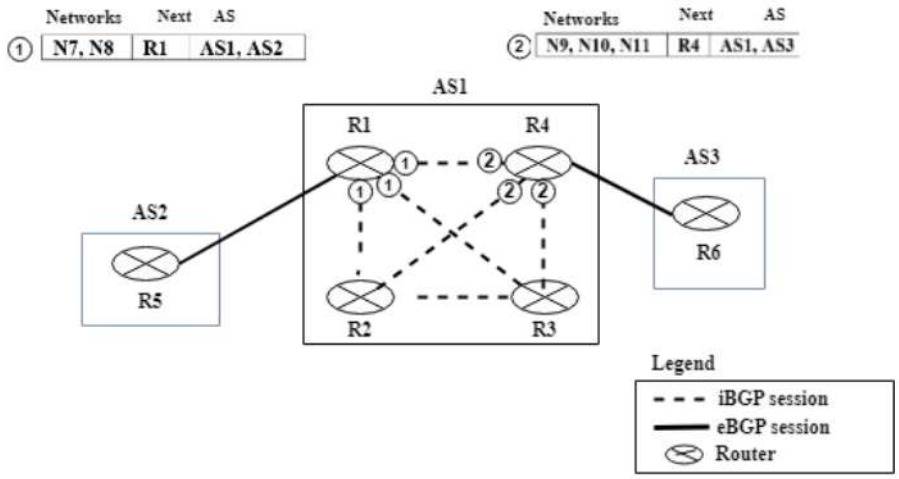


Fig. 8.17: Combination of eBGP and iBGP sessions for the sample Internet

After a number of messages are exchanged between all the routers, the finalized BGP path tables are created as shown in the Fig. 8.18.

Networks	Next	Path
N7, N8	R5	AS1, AS2
N9, N10, N11	R4	AS1, AS3

Path Table for R1

Networks	Next	Path
N7,N8	R1	AS1, AS2
N9,N10,N11	R1	AS1, AS3

Path Table for R2

Networks	Next	Path
N9,N10,N11	R4	AS1, AS3
N7,N8	R2	AS1, AS2

Path Table for R3

Networks	Next	Path
N7, N8	R1	AS1, AS2
N9, N10, N11	R6	AS1, AS3

Path Table for R4

Networks	Next	Path
N1, N2, N3, N4	R1	AS1, AS2
N9, N10, N11	R1	AS2, AS1, AS3

Path Table for R5

Networks	Next	Path
N7, N8	R4	AS1, AS2, AS3
N1, N2, N3. N4	R4	AS1, AS3

Path Table for R6

Fig. 8.18: Finalized BGP path tables

8.5.6.3 Performance

The performance of BGP is as good as RIP. The BGP peers exchange a number of message for creating the forwarding tables in order to reduce loops and count-to-infinity problems.

8.6 Self-Assessment Questions

Q1. What is Network Address Translation ? Mention its use in networking. (6 marks, L4)

Q2. Why was IPv6 developed and what are its key features ? (10 marks, L4)

Q3. What is a Routing Algorithm and what are its goals ? (6 marks, L3)

Q4. List the different categories of routing algorithms and briefly explain each of them. (12 marks, L3)

Q5. Compare Distance Vector and Link State Routing. (10 marks, L4)

Q6. Explain the role of Hop count in routing algorithms. (4 marks, L3)

Q7. Explain RIP with suitable diagrams (8 marks, L3)

Q8. Briefly explain the OSPF routing algorithm (8 marks, L3)

Q9. Briefly explain the Border Gateway Protocol (8 marks, L3)

8.7 Multiple-Choice Questions

Q1. The main purpose of NAT is to, [1 mark, L1]

A. Establish connection between two hosts
B. Convert private IP address to public IP address
C. To encrypt network traffic
D. None of the above

Q2. NAT operates at _____ layer, [1 mark, L1]

A. Physical

B. Link

C. Network

D. Application

Q3. Which of the statement is true about NAT ?, [1 mark, L1]

A. It hides the network from external access

B. It converts domain names to physical address

C. It converts domain names to IP address

D. It allows devices with private IP address to access Internet with a public IP address

Q4. The main goal of developing IPv6 is, [1 mark, L1]

A. To increase data transmission speed

B. To improve network security

C. To increase the number of available IP addresses

D. None of the above

Q5. The total number of bits in the IP address is ___ [1 mark, L1]

A. 32

B. 64

C. 128

D. None of the above

Q6. Which of the following routing algorithms is used in the Internet Protocol (IP)? [1 mark, L1]

A. Bellman-Ford

B. Dijkstra's

C. Distance vector

D. Link-state

Q7. In a network with N routers, there will be _____ least-cost paths between one router to all other routers. [1 mark, L1]

A. N - 1

B. N(N-1)

C. N^2

D. None of the above

Q8. _____ table is used by the router to find the next hop for a packet. [1 mark, L1]

A. Switching

B. ARP

C. NAT

D. Routing

Q9. _____ algorithm is used to find the shortest path for a packet [1 mark, L1]

A. Bellman-Ford

B. Distance vector

C. Link-state routing

D. All of the above.

Q10. RIP is a _____ type of routing protocol [1 mark, L1]

A. Distance vector
B. Link State
C. Path Vector
D. None of the above

Q.11 The metric used by RIP to find the best path is ____ [1 mark, L1]

A. Delay
B. Bandwidth
C. Hop count
D. None of the above

Q12. The maximum number of hop count to prevent routing loops in RIP is ___ [1 mark, L1]

A. 10
B. 15
C. 16
D. 20

Q13. OSPF is a ____ type of a routing protocol [1 mark, L1].

A. Distance vector
B. Link State
C. Path Vector
D. None of the above

Q14. In OSPF, the Hello packet is used to, [1 mark, L1]

A. To broadcast routing updates
B. To get the state of the link
C. To build an LSDB
D. To establish neighbour relationship

8.8 Keys to Multiple-Choice Questions

Q1. To convert private IP address to public IP address (B)
Q2. Network (C)
Q3. It allows devices with private IP address to access the Internet with a shared public IP address (D)
Q4. To increase the number of available IP addresses (C)
Q5. 128 (C)
Q6. Distance Vector (C)
Q7. N-1 (A)
Q8. Routing (D)
Q9. All the algorithms (D)
Q10. Distance Vector (A)
Q11. Hop count (C)
Q12. 15 (B)
Q13. Link state (B)

Q14. To establish neighbour relationship (D)

8.9 Summary of the Unit

This unit covers three important topics: Network Address Translation, Internet Protocol Version 6 and Routing algorithms. The first section of this unit discussed the NAT process which has various benefits such as conserving public IP addresses by associating a single public IP address for a set of private addresses for devices in a private network and hence delay the transition to IPv6. NAT devices can act as a firewall and hide private IP addresses within the network. In the second section, the features of IPv6 and the challenges in the transitioning of IPv4 to IPv6 were discussed. In the following sections, routing algorithms were discussed. The main goal of routing is to find the best route for a packet from its source to its destination. The best route can be in terms of delay, bandwidth, and load.

Routing algorithms are designed based on two popular graph-based algorithms to find the shortest path or minimum cost between source and destination nodes: Bellman-Ford and Dijkstra. Two important algorithms Link-state and Distance vector designed based on these algorithms are discussed. So routing determines the best path for data to travel between networks, while forwarding is the process of transmitting data along that path. Both processes are essential for the proper functioning of computer networks.

8.10 Recommended Learning Resources

[1] James F Kurose and Keith W Ross, Computer Networking, A Top-Down Approach, Sixth Edition, Pearson,2017.

[2] Behrouz A Forouzan, Data and Communications and Networking, Fifth Edition, McGraw Hill, Indian Edition

8.11 References

[1] https://users.encs.concordia.ca/~dongyu/ELEC6851/lec12.pdf

[2] https://www.gatevidyalay.com/distance-vector-routing-routing-algorithms/

[3] https://www.baeldung.com/cs/routing-vs-forwarding-tables

[4] https://www.thecoldwire.com/what-is-the-difference-between-routing-and-forwarding/

[5] https://www.quora.com/What-is-the-difference-between-a-routing-table-and-a-forwarding-table

MCQs:

[1] https://t4tutorials.com/router-mcqs-networking-devices/

[2] ChatGPT 3.5