NETWORK LAYER

Services :
The **services** which are offered by the network layer protocol are as follows:

1. **Packetizing –**
   The process of encapsulating the data received from upper layers of the network(also called as payload) in a network layer packet at the source and decapsulating the payload from the network layer packet at the destination is known as packetizing.
   The source host adds a header that contains the source and destination address and some other relevant information required by the network layer protocol to the payload received from the upper layer protocol, and delivers the packet to the data link layer.

   The destination host receives the network layer packet from its data link layer, decapsulates the packet, and delivers the payload to the corresponding upper layer protocol. The routers in the path are not allowed to change either the source or the destination address. The routers in the path are not allowed to decapsulate the packets they receive unless they need to be fragmented.

2. **Routing and Forwarding –**
   These are two other services offered by the network layer. In a network, there are a number of routes available from the source to the destination. The network layer specifies has some strategies which find out the best possible route. This process is referred to as routing. There are a number of routing protocols which are used in this process and they should be run to help the routers coordinate with each other and help in establishing communication throughout the network.
   Forwarding is simply defined as the action applied by each router when a packet arrives at one of its interfaces. When a router receives a packet from one of its attached networks, it needs to forward the packet to another attached network (unicast routing) or to some attached networks(in case of multicast routing).

Some of the other **services which are expected** from the network layer are:

1. **Error Control –**
   Although it can be implemented in the network layer, but it is usually not preferred because the data packet in a network layer maybe fragmented at each router, which makes error checking inefficient in the network layer.

2. **Flow Control –**
   It regulates the amount of data a source can send without overloading the receiver. If the source produces a data at a very faster rate than the receiver can consume it, the receiver will be overloaded with data. To control the flow of data, the receiver should send a feedback to the sender to inform the latter that it is overloaded with data.
   There is a lack of flow control in the design of the network layer. It does not directly provide any flow control. The datagrams are sent by the sender when they are ready, without any attention to the readiness of the receiver.

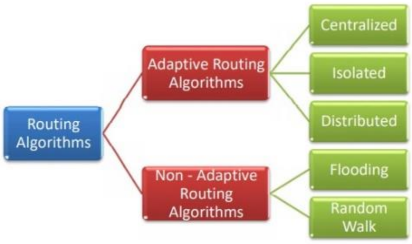
3. **Congestion Control –**
   Congestion occurs when the number of datagrams sent by source is beyond the capacity of network or routers. This is another issue in the network layer protocol. If congestion continues, sometimes a situation may arrive where the system collapses and no datagrams are delivered. Although congestion control is indirectly implemented in network layer, but still there is a lack of congestion control in the network layer.

Routing algorithm

A routing algorithm is a procedure that lays down the route or path to transfer data packets from source to the destination. They help in directing Internet traffic efficiently. After a data packet leaves its source, it can choose among the many different paths to reach its destination. Routing algorithm mathematically computes the best path, i.e. “least – cost path” that the packet can be routed through.

Types of Routing Algorithms

Routing algorithms can be broadly categorized into two types, adaptive and nonadaptive routing algorithms. They can be further categorized as shown in the following diagram −



Adaptive Routing Algorithms

Adaptive routing algorithms, also known as dynamic routing algorithms, makes routing decisions dynamically depending on the network conditions. It constructs the routing table depending upon the network traffic and topology. They try to compute the optimized route depending upon the hop count, transit time and distance.

The three popular types of adaptive routing algorithms are −

- **Centralized algorithm** − It finds the least-cost path between source and destination nodes by using global knowledge about the network. So, it is also known as global routing algorithm.
- **Isolated algorithm** − This algorithm procures the routing information by using local information instead of gathering information from other nodes.
- **Distributed algorithm** − This is a decentralized algorithm that computes the least-cost path between source and destination iteratively in a distributed manner.

Non – Adaptive Routing Algorithms

Non-adaptive Routing algorithms, also known as static routing algorithms, construct a static routing table to determine the path through which packets are to be sent. The static routing table is constructed based upon the routing information stored in the routers when the network is booted up.

The two types of non – adaptive routing algorithms are −

- **Flooding** − In flooding, when a data packet arrives at a router, it is sent to all the outgoing links except the one it has arrived on. Flooding may be uncontrolled, controlled or selective flooding.
- **Random walks** − This is a probabilistic algorithm where a data packet is sent by the router to any one of its neighbours randomly.

Distance Vector Routing Algorithm

- **The Distance vector algorithm is iterative, asynchronous and distributed.**
  - **Distributed:** It is distributed in that each node receives information from one or more of its directly attached neighbors, performs calculation and then distributes the result back to its neighbors.
  - **Iterative:** It is iterative in that its process continues until no more information is available to be exchanged between neighbors.
  - **Asynchronous:** It does not require that all of its nodes operate in the lock step with each other.

- The Distance vector algorithm is a dynamic algorithm.
- It is mainly used in ARPANET, and RIP.
- Each router maintains a distance table known as **Vector**.

Distance Vector Routing Algorithm

Let $d_x(y)$ be the cost of the least-cost path from node x to node y. The least costs are related by Bellman-Ford equation,

$$d_x(y) = \min_v\{c(x,v) + d_v(y)\}$$

Where the minv is the equation taken for all x neighbors. After traveling from x to v, if we consider the least-cost path from v to y, the path cost will be $c(x,v)+d_v(y)$. The least cost from x to y is the minimum of $c(x,v)+d_v(y)$ taken over all neighbors.

Three Keys to understand the working of Distance Vector Routing Algorithm:

- **Knowledge about the whole network:** Each router shares its knowledge through the entire network. The Router sends its collected knowledge about the network to its neighbors.
- **Routing only to neighbors:** The router sends its knowledge about the network to only those routers which have direct links. The router sends whatever it has about the network through the ports. The information is received by the router and uses the information to update its own routing table.
- **Information sharing at regular intervals:** Within 30 seconds, the router sends the information to the neighboring routers.

With the Distance Vector Routing algorithm, the node x contains the following routing information:

- For each neighbor v, the cost c(x,v) is the path cost from x to directly attached neighbor, v.
- The distance vector x, i.e., D_x = [$D_x(y)$: y in N], containing its cost to all destinations, y, in N.
- The distance vector of each of its neighbors, i.e., D_v = [$D_v(y)$: y in N] for each neighbor v of x.

Distance vector routing is an asynchronous algorithm in which node x sends the copy of its distance vector to all its neighbors. When node x receives the new distance vector from one of its neighboring vector, v, it saves the distance vector of v and uses the Bellman-Ford equation to update its own distance vector. The equation is given below:

$$d_x(y) = \min_v\{ c(x,v) + d_v(y)\} \quad \text{for each node } y \text{ in } N$$

The node x has updated its own distance vector table by using the above equation and sends its updated table to all its neighbors so that they can update their own distance vectors.

Algorithm

```
At each node x,
```

Initialization

```
for all destinations y in N:
Dx(y) = c(x,y)     // If y is not a neighbor then c(x,y) = ∞
for each neighbor w
Dw(y) = ?     for all destination y in N.
for each neighbor w
send distance vector Dx = [ Dx(y)  : y in N ] to w
```

loop

```
  wait (until I receive any distance vector from some neighbor w)
  for each y in N:
  Dx(y) = minv{c(x,v)+Dv(y)}
If Dx(y) is changed for any destination y
Send distance vector Dx = [ Dx(y) : y in N ] to all neighbors
```

forever

Link State Routing

Link state routing is a technique in which each router shares the knowledge of its neighborhood with every other router in the internetwork.

The three keys to understand the Link State Routing algorithm:

- **Knowledge about the neighborhood:** Instead of sending its routing table, a router sends the information about its neighborhood only. A router broadcast its identities and cost of the directly attached links to other routers.
- **Flooding:** Each router sends the information to every other router on the internetwork except its neighbors. This process is known as Flooding. Every router that receives the packet sends the copies to all its neighbors. Finally, each and every router receives a copy of the same information.
- **Information sharing:** A router sends the information to every other router only when the change occurs in the information.

Link State Routing has two phases:

Reliable Flooding

- **Initial state:** Each node knows the cost of its neighbors.
- **Final state:** Each node knows the entire graph.

Route Calculation

Each node uses Dijkstra's algorithm on the graph to calculate the optimal routes to all nodes.

- The Link state routing algorithm is also known as Dijkstra's algorithm which is used to find the shortest path from one node to every other node in the network.
- The Dijkstra's algorithm is an iterative, and it has the property that after k^{th} iteration of the algorithm, the least cost paths are well known for k destination nodes.

Let's describe some notations:

- **c(i , j):** Link cost from node i to node j. If i and j nodes are not directly linked, then c(i , j) = ∞.
- **D(v):** It defines the cost of the path from source code to destination v that has the least cost currently.
- **P(v):** It defines the previous node (neighbor of v) along with current least cost path from source to v.
- **N:** It is the total number of nodes available in the network.

Algorithm

Initialization

```
N = {A}      // A is a root node.
for all nodes v
if v adjacent to A
then D(v) = c(A,v)
else D(v) = infinity
```

loop

```
find w not in N such that D(w) is a minimum.
Add w to N
Update D(v) for all v adjacent to w and not in N:
D(v) = min(D(v) , D(w) + c(w,v))
Until all nodes in N
```

Path-vector routing protocol

A **path-vector routing protocol** is a network routing protocol which maintains the path information that gets updated dynamically. Updates that have looped through the network and returned to the same node are easily detected and discarded. This algorithm is sometimes used in Bellman–Ford routing algorithms to avoid "Count to Infinity" problems.

It is different from the distance vector routing and link state routing. Each entry in the routing table contains the destination network, the next router and the path to reach the destination.

It has three phases:

1. Initiation
2. Sharing
3. Updating

Intradomain and Interdomain Routing

S.No	Intradomain Routing	Interdomain Routing
1.	Routing algorithm works only within domains.	Routing algorithm works within and between domains.
2.	It need to know only about other routers within their domain.	It need to know only about other routers

		within and between their domain.
3.	Protocols used in intradomain routing are known as **Interior-gateway protocols**.	Protocols used in interdomain routing are known as **Exterior-gateway protocols**.
4.	In this Routing, routing takes place within an autonomous network.	In this Routing, routing takes place between the autonomous networks.
5.	Intradomain routing protocols ignores the internet outside the AS(autonomous system).	Interdomain routing protocol assumes that the internet contains the collection of interconnected AS(autonomous systems).
6.	Some Popular Protocols of this routing are RIP(resource information protocol) and OSPF(open shortest path first).	Popular Protocols of this routing is BGP(Border Gateway Protocol) used to connect two or more AS(autonomous system).