## Routing:

- **Routing** is the process of selecting best paths in a network.
- Routing is performed for many kinds of networks, including the telephone network (circuit switching), electronic data networks (such as theInternet), and transportation networks
- In packet switching networks, routing directs packet forwarding (the transit of logically addressed network packets from their source toward their ultimate destination) through intermediate nodes. Intermediate nodes are typically network hardware devices such as routers, bridges, gateways, firewalls, or switches
- The routing process usually directs forwarding on the basis of routing tables, which maintain a record of the routes to various network destinations
- Thus, constructing routing tables, which are held in the router's memory, is very important for efficient routing. Most routing algorithms use only one network path at a time. Multipath routing techniques enable the use of multiple alternative paths.

### Design Goals :

Routing algorithms often have one or more of the following design goals:

- ***Optimality* -** the capability of the routing algorithm to select the best route, depending on metrics and metric weightings used in the calculation. For example, one algorithm may use a number of hops and delays, but may weight delay more heavily in the calculation.
- ***Simplicity and low overhead* -** efficient routing algorithm functionality with a minimum of software and utilization overhead. Particularly important when routing algorithm software must run on a computer with limited physical resources.
- ***Robustness and stability* -** routing algorithm should perform correctly in the face of unusual or unforeseen circumstances, such as hardware failures, high load conditions, and incorrect implementations. Because of their locations at network junctions, failures can cause extensive problems.
- ***Rapid convergence* -** Convergence is the process of agreement, by all routers, on optimal routes. When a network event causes changes in router availability, recalculations are need to restablish networks. Routing algorithms that converge slowly can cause routing loops or network outages.
- ***Flexibility* -** routing algorithm should quickly and accurately adapt to a variety of network circumstances. Changes of consequence include router availability, changes in network bandwidth, queue size, and network delay.

### Desirable Characteristics of Routing AlgorithM:

- Correctness
- Simplicity

- Robustness
- Stability (convergence to an equilibrium)
- Optimality
- Fairness

Routing Algorithm Classification:

Routing tables can contain directly connected, manually configured static routes and routes learned dynamically using a routing protocol. Network professionals must understand when to use static or dynamic routing.

**Non-adaptive (static) algorithms:**

– Route choices computed in advance and downloaded to routers at network

boot time

Static routing has several primary uses, including:

- Providing ease of routing table maintenance in smaller networks that are not expected to grow significantly.
- Routing to and from a stub network, which is a network with only one default route out and no knowledge of any remote networks.
- Accessing a single default route (which is used to represent a path to any network that does not have a more specific match with another route in the routing table).
- Static routing is easy to implement in a small network. Static routes stay the same, which makes them fairly easy to troubleshoot. Static routes do not send update messages and, therefore, require very little overhead.

**Advantages**

- Easy to implement in a small network
- Very secure. No advertisements are sent, unlike with dynamic routing protocols.
- It is very predictable, as the route to the destination is always the same
- No routing algorithm or update mechanisms are required. Therefore, extra resources (CPU and memory) are not required.

**Disadvantage:**

- Suitable for simple topologies or for special purposes such as a default static route.
- Configuration complexity increases dramatically as the network grows. Managing the static configurations in large networks can become time consuming.
- If a link fails, a static route cannot reroute traffic. Therefore, manual intervention is required to re-route traffic.

**• Adaptive (dynamic) algorithms:**

- Adapt routing decisions in response to topology and traffic changes.
- Dynamic routing protocols work well in any type of network consisting of several routers. They are scalable and automatically determine better routes if there is a change in the topology. Although there is more to the configuration of dynamic routing protocols, they are simpler to configure in a large network.

**Advantage:**

- Suitable in all topologies where multiple routers are required.
- Generally independent of the network size.
- Automatically adapts topology to reroute traffic if possible.

**Disadvantage:**

- Can be more complex to initially implement.
- Less secure due to the broadcast and multicast routing updates. Additional configuration settings such as passive interfaces and routing protocol authentication are required to increase security.
- Route depends on the current topology.
- Requires additional resources such as CPU, memory, and link bandwidth.

**Types Of Routing Algorithm:**

**Fixed Path Routing**

- Fixed path routing is the simplest approach to finding a light path.
- The same fixed route for a given source and destination pair is always used.
- Typically this path is computed ahead of time using a shortest path algorithm, such as Dijkstra's Algorithm.
- While this approach is very simple, the performance is usually not sufficient. If resources along the fixed path are in use, future connection requests will be blocked even though other paths may exist.
- The Shortest path routing algorithm is an example of a Fixed Path Routing .
- Fixed path routing is simple to implement and thus has a lower computational complexity.

- Since it imposes a strict restriction on routing ,it can result in a higher request blocking probability in the network
- It doesn't have a fault tolerant capability.

**Shortest Path Routing:**

- **Shortest path routing** refers to the process of finding paths through a network that have a minimum of distance or other cost metric
- in shortest path routing ,the topology communications network is represented using a directed weighted graph
- the node in the graph represent the switching elements and the directed arcs in the graph represents the communication links between switching elements.
- Each arc has a weight that represents the cost of sending packet between two nodes in a particular direction.
- This cost is generally a positive value that could be function of distance,band width , average traffic , communication cost, router processing speed, measured delay througput ,error rate etc.
- The objective of shortest path routing is to find a path between two nodes that has the smallest total cost , where the total cost of a path is the sum of the arc costs in path.
- Shortest path algorithm can be divided into two classes ie distance vector and link state.

Example with figure……

**Flooding:**

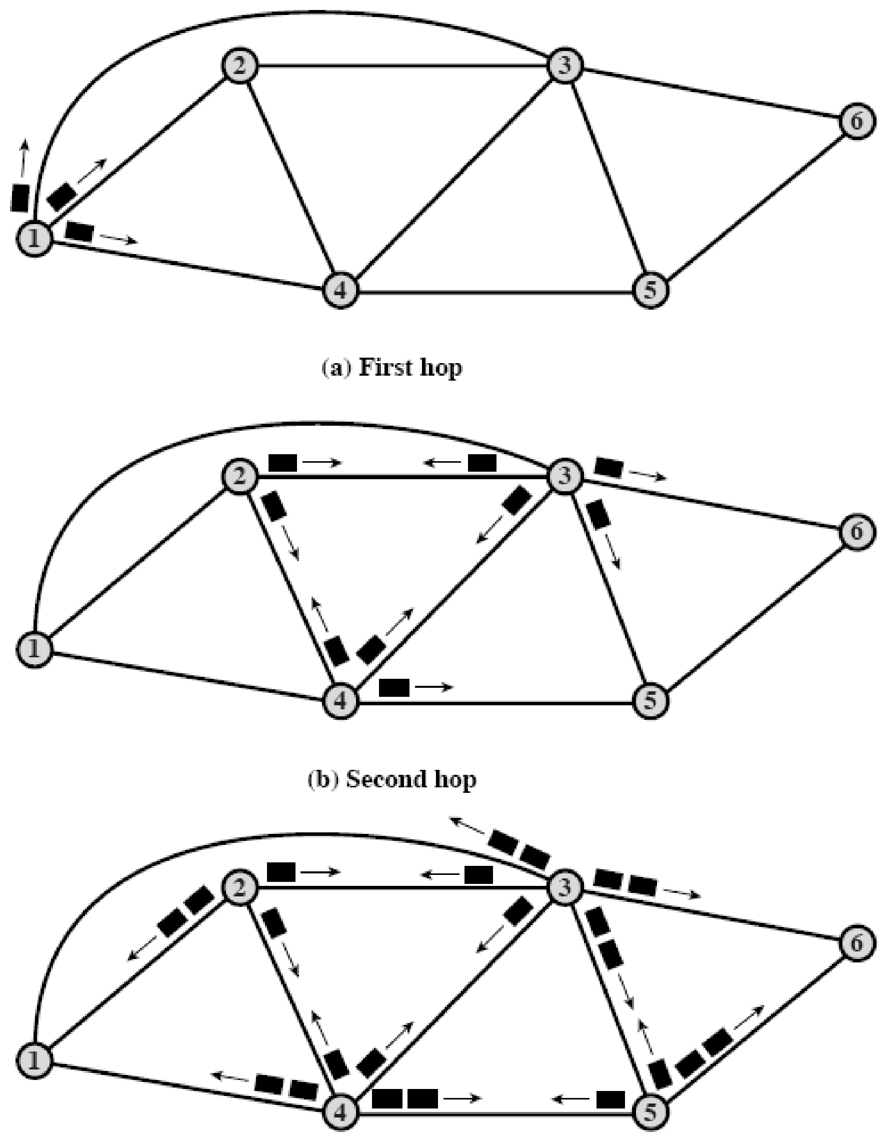
- Flooding is the static routing algorithm. In this algorithm, every incoming packet is sent on all outgoing lines except the line on which it has arrived.
- Traffic grows very quickly when every node floods the packets so eventually multiple copies arrive at the destination.
- Since flooding naturally utilizes every path through the network, it will also use the shortest path.
- This algorithm is very simple to implement
- Messages can become duplicated in the network further increasing the load on the networks bandwidth as well as requiring an increase in processing complexity to disregard duplicate messages.
- There are generally two types of flooding available, **Uncontrolled Flooding and Controlled Flooding.**
- In Uncontrolled Flooding all nodes have neighbours and route packets indefinitely. More than two neighbours creates a broadcast storm.

- Controlled Flooding has its own two algorithms to make it reliable, SNCF (Sequence Number Controlled Flooding) and RPF (Reverse Path Flooding).
- In SNCF, the node attaches its own address and sequence number to the packet, since every node has a memory of addresses and sequence numbers. If it receives a packet in memory, it drops it immediately
- In RPF, the node will only send the packet if it is arrived on predecessors otherwise discard the packet that is either already arrived or will arrive from predecessor
- One major problem of this algorithm is that it generates a large number of duplicate packets on the network.
- Several measures are takes to stop the duplication of packets. These are:

1. One solution is to include a hop counter in the header of each packet. This counter is decremented at each hop along the path. When this counter reaches zero the packet is discarded. Ideally, the hop counter should become zero at the destination hop, indicating that there are no more intermediate hops and destination is reached. This requires the knowledge of exact number of hops from a source to destination.

2 . Another technique is to keep the track of the packed that have been flooded, to avoid sending them a second time. For this, the source router put a sequence number in each packet it receives from its hosts. Each router then needs a list per source router telling which sequence numbers originating at that source have already been seen. If an incoming packet is on the list, it is not flooded.

3. Another solution is to use **selective flooding.** In selective flooding the routers do not send every incoming packet out on every output line. Instead packet is sent only on those lines which are approximately going in the right direction.




(a) First hop

(b) Second hop

(c) Third hop

**Distance Vector Routing:**

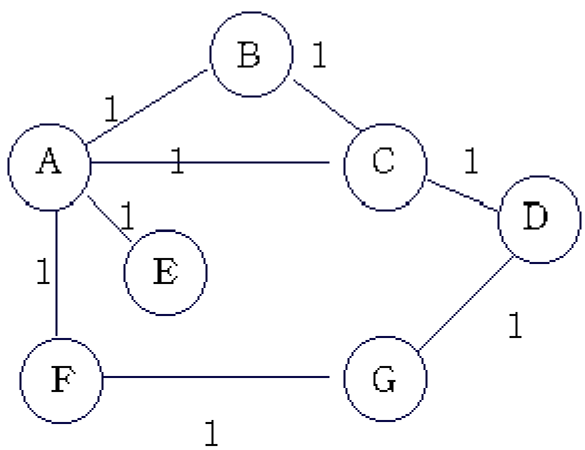
- Distance-vector routing protocols use the Bellman–Ford algorithm, Ford–Fulkerson algorithm
- Determine the direction (vector) and distance (hop count) to any link in the internet
- A typical distance vector routing protocol uses a routing algorithm in which routers periodically send routing updates to all neighbors by broadcasting their entire route tables
- Table contains two parts
    - The preferred out going line to use for that destination

- An estimate of the time or distance to that destination

➢ Each node constructs a one-dimensional array containing the "distances"(costs) to all other nodes and distributes that vector to its immediate neighbors.

1. The starting assumption for distance-vector routing is that each node knows the cost of the link to each of its directly connected neighbors.
2. A link that is down is assigned an infinite cost.

Example.




| **Information Stored at Node** | **Distance to Reach Node** A | **B** | **C** | **D** | **E** | **F** | **G** |
|---|---|---|---|---|---|---|---|
| **A** | 0 | 1 | 1 | ? | 1 | 1 | ? |
| **B** | 1 | 0 | 1 | ? | ? | ? | ? |
| **C** | 1 | 1 | 0 | 1 | ? | ? | ? |
| **D** | ? | ? | 1 | 0 | ? | ? | 1 |
| **E** | 1 | ? | ? | ? | 0 | ? | ? |
| **F** | 1 | ? | ? | ? | ? | 0 | 1 |
| **G** | ? | ? | ? | 1 | ? | 1 | 0 |

**Table 1. Initial distances stored at each node(global view).**

We can represent each node's knowledge about the distances to all other nodes as a table like the one given in Table 1.

Note that each node only knows the information in one row of the table.

1. Every node sends a message to its directly connected neighbors containing its personal list of distance. ( for example, **A** sends its information to its neighbors **B,C,E**, and **F**. )
2. If any of the recipients of the information from **A** find that **A** is advertising a path shorter than the one they currently know about, they update their list to give the new path length and note that they should send packets for that destination through **A**. ( node **B** learns from **A** that node **E** can be reached at a cost of 1; **B** also knows it can reach **A** at a cost of 1, so it adds these to get the cost of reaching **E** by means of **A**. **B** records that it can reach **E** at a cost of 2 by going through **A**.)
3. After every node has exchanged a few updates with its directly connected neighbors, all nodes will know the least-cost path to all the other nodes.
4. In addition to updating their list of distances when they receive updates, the nodes need to keep track of which node told them about the path that they used to calculate the cost, so that they can create their forwarding table. ( for example, **B** knows that it was **A** who said " I can reach **E** in one hop" and so **B** puts an entry in its table that says " To reach **E**, use the link to **A**.)

| **Information** | **Distance to Reach Node** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Stored at Node** | **A** | **B** | **C** | **D** | **E** | **F** | **G** |
| **A** | 0 | 1 | 1 | 2 | 1 | 1 | 2 |
| **B** | 1 | 0 | 1 | 2 | 2 | 2 | 3 |
| **C** | 1 | 1 | 0 | 1 | 2 | 2 | 2 |
| **D** | 2 | 2 | 1 | 0 | 3 | 2 | 1 |
| **E** | 1 | 2 | 2 | 3 | 0 | 2 | 3 |
| **F** | 1 | 2 | 2 | 2 | 2 | 0 | 1 |
| **G** | 2 | 3 | 2 | 1 | 3 | 1 | 0 |

**Table 2. final distances stored at each node ( global view).**

In practice, each node's forwarding table consists of a set of triples of the form:

( Destination, Cost, NextHop).

For example, Table 3 shows the complete routing table maintained at node B for the network in figure1.

| Destination | Cost | NextHop |
|---|---|---|
| **A** | 1 | A |
| **C** | 1 | C |
| **D** | 2 | C |
| **E** | 2 | A |
| **F** | 2 | A |
| **G** | 3 | A |

**Table 3. Routing table maintained at node B.**

**Link State Routing Algorithm:**

- Link state algorithm enables router to focus on their own links and interfaces
- Any one router on a network will only have direct knowledge of the router that are directly connected to it.
- Because of a link state protocol only maintains routing information for its direct interfaces , the routing table contains much less information that that of a distance vector protocol that might have information for multiple router
- The router simply needs to know which one of its direct interface will get the information where it needs to go the quickest, the next router in line will repeat the process until the information reaches its destination
- Sends trigger update only when a network change has occurred and send a periodic updates at along time interval such as every 30 minutes
- OSPF used in internet uses the link state algorithm
- When a route or link changes the device that detected the change creates a link state advertisement (LSA) concerning that link. The LSA is then transmitted to all neighboring devices. Each routing device take a copy of the LSA, updates its link state database and forward the LSA to all neighboring devices
- Link state router finds the best path to destination by applying Dijkstra Shortest Path (SPF) algorithm.
- Link-state protocols maintain three separate tables:

  - **Neighbor table** – contains a list of all neighbors, and the interface each neighbor is connected off of. Neighbors are formed by sending Hello packets.

- **Topology table** – otherwise known as the "link-state" table, contains a map of all links within an area, including each link's status.

- **Routing table** – contains the best routes to each particular destination

## Difference between Distance Vector and Link State Routing:

- Distance vector protocols are used in small networks, and it has a limited number of hops, whereas Link state protocol can be used in larger networks, and it has unlimited number of hops
- Distance vector protocol has a high convergence time, but in link state, convergence time is low.
- Distance vector protocol periodically advertise updates, but link state advertises only new changes in a network.
- Distance vector protocol advertises only the directly connected routers and full routing tables, but link state protocols only advertise the updates, and flood the advertisement.
- In distance vector protocol, loop is a problem, and it uses split horizon, route poisoning and hold down as loop preventing techniques, but link state has no loop problems.
- Distance vector maintains only one routing table but link state maintains three tables (neighbor table, topology table and routing table)
- Distance vector is simple to implement and configure where as link state is more complex

## Advantages of Distance Vector routing:

- It is simpler to configure than Link State
- It is simpler to maintain than Link State

## Disadvantages of Distance Vector routing:

- It is slower to converge than Link State
- It is at risk from the count-to-infinity problem
- It creates more traffic than Link State since a hop count change must be propagated to all routers and processed on each router. Hop count updates take place on a periodic basis, even if there are no changes in the network topology, so bandwidth-wasting broadcasts still occur.

- For larger networks, Distance Vector routing results in larger routing tables than Link State since each router must know about all other routers. This can also lead to congestion on WAN links. RIP announces host or default routes by default.

The **count-to-infinity problem** happens when a router is unableto reach an adjoining network. A second router, 1 hop away from the first router, thinks that the unreachable network is 2 hops away. Meanwhile, the first router then updates its records to say it is 3 hops away from the unreachable network based on the fact it is 1 hop from the second router, which says it is 2 hops from the unreachable network. The routers continue incrementing their hop count until the maximum (15), "infinity", is reached.

## Advantage and Disadvantage of Link State Routing Algorithm

- Link-state protocols use cost metrics to choose paths through the network. The cost metric reflects the capacity of the links on those paths.
- Link-state protocols use triggered updates and LSA floods to immediately report changes in the network topology to all routers in the network. This leads to fast convergence times.
- Each router has a complete and synchronized picture of the network. Therefore, it is very difficult for routing loops to occur.
- Routers use the latest information to make the best routing decisions.
- The link-state database sizes can be minimized with careful network design. This leads to smaller Dijkstra calculations and faster convergence.
- Every router, at the very least, maps the topology of its own area of the network. This attribute helps to troubleshoot problems that can occur.
- Link-state protocols support CIDR and VLSM.

The following are some disadvantages of link-state routing protocols:

- They require more memory and processor power than distance vector protocols. This makes it expensive to use for organizations with small budgets and legacy hardware.
- They require strict hierarchical network design, so that a network can be broken into smaller areas to reduce the size of the topology tables.
- They require an administrator who understands the protocols well.
- They flood the network with LSAs during the initial discovery process. This process can significantly decrease the capability of the network to transport data. It can noticeably degrade the network performance.