Routing techniques
There are two techniques of routing:
1. Non-adaptive (Static routing)
2. Adaptive (Dynamic routing)

Static Routing
In this technique, the router is configured manually. The choice of the route to use to get from i to j is computed in advance, offline and downloaded to the router when the network is booked. The manual maintenance of the routing table for the large network could require a lot of administrative time.
Advantages:
- Easily implemented in a small network.
- It is very predictable, as the route to the destination is always the same.

Disadvantages:
- Only suitable for small network cannot work for large network.
- Complexity increases when the network becomes very complex and very time taking.
- It cannot support complex routing algorithms.

Dynamic Routing
Dynamic routing makes it possible to avoid the configuration of the static routes. The router decision changes to reflect changes in the topology and usually the traffic as well.
Advantages:
- Easily used in large network
- It supports more complex routing algorithms

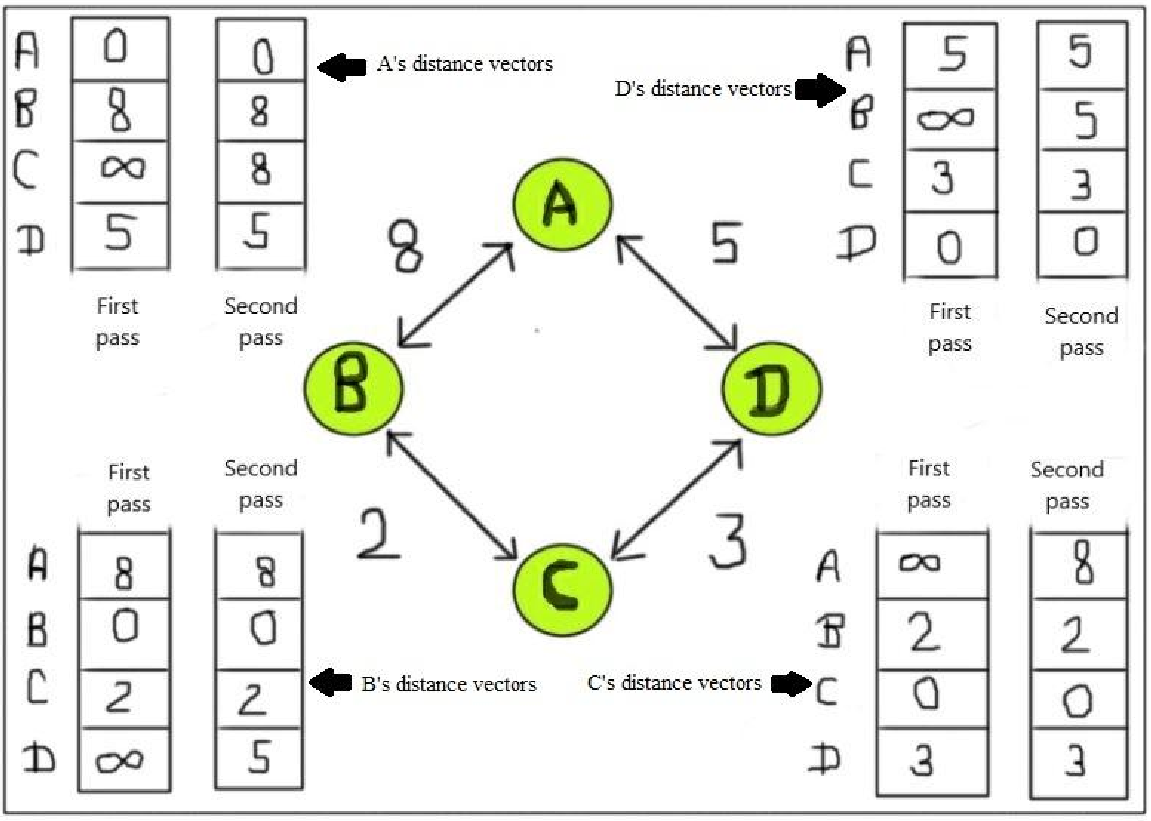
Disadvantages:
- Less secure than static routing because of a multicast routing change.
- Additional configuration setting is required such as a routing protocol, passive interfaces to increase security.

Distance Vector Algorithm

The distance vector algorithm mainly does the following -

1. All routers inform each other about their distance vector periodically. Every router initially has information only about its neighbours in the first pass of the distance vector table.
2. Every router sends and receives the latest distance vector, from its neighbouring routers.
3. In case any router's distance vector changes, it informs its neighbours accordingly.
4. The distance vector sent by each router contains the least costly path to reach every other.

Let's see how the entire process works and how the distance vector table updates for each node in a network. For this, consider four routers in a network, named A, B, C, and D, respectively.



We first begin with router A's distance vector. As we can see in the image below, router A initially contains path information only about its neighboring routers. So, the cost of routers that are not its neighbors is set as infinity. In this first pass of the distance vector for router A, we have a path to reach routers B and D, but not C. Parallelly, we repeat this same process for all routers in the network.

Now we check the distance vector for router B. Router B's neighbours are router A and router C. Therefore, B's distance vector in the initial pass contains path costs to reach router A and router C. Next, we do the same check for router C. C's neighbours are router B and router D. Thus, C's distance vector initially only knows the path costs from router C to routers B and D, respectively. Lastly, we check the distance vector for router D. D's neighbours are router A and C. So, we include their path costs in router D's initial distance vector table.

Now, we move on to the second iteration. We start again with router A. Here, using distance vector tables of A's neighbours B and D from the previous iteration, we find additional paths. From the previous iteration, we see from router B's distance vector that we can reach router C from A at a cost of 10. We use B as the connecting router to reach C (here, we can call B as the intermediate hop for A to reach C). Likewise, from router D's distance vector, we can reach C from A using D as an intermediate hop. This gives us a path cost of eight. Since the path from A to D and then to C costs less than the route to C from A using B, we choose router D to reach router C from A. Now, we update router A's distance vector table. A now informs its neighbours about changes in its distance vector.

Next, we do the same for router B. Using the distance vectors of B's neighbours from previous iterations, we find that we can reach router D from B using router A or router C, at a cost of 13 and 5, respectively. Hence, we choose router C to reach router D from B and update B's distance vector. Now, we check router C's distance vector. We can reach router A from router C using router B or D, at a cost of 10 and 8, respectively. So, we choose the path using router D and update router C's distance vector.

Lastly, we check router D's distance vector. Using the information from D's neighbours and their distance vector, we have a path to reach router B from D, using router A or C. Since using C us an intermediate hop costs less to reach router B from D, we choose this path. Finally, we update router D's distance vector. This way, the distance vector algorithm finds the least-cost paths from one router to another in a network infrastructure.

Link State Routing

In the Link - State Routing Protocol, the router attempts to construct its own internal map of the network topology. It provides the information about whether the link to reach the router is active or not.

Every router will create something called Link state packets.

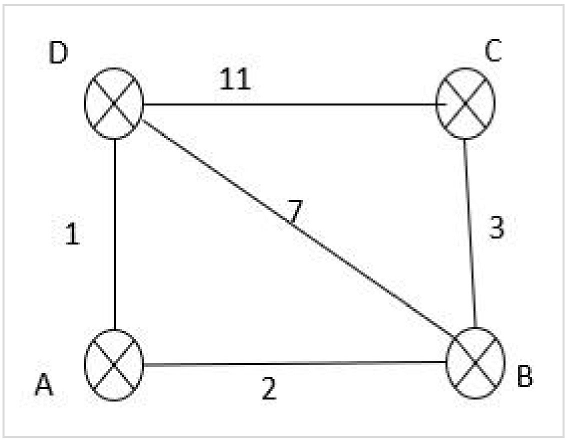
In the first round every node creates link state packets with the help of “Hello packets”.

Step 1 − Prepare the link state packet at every router.

D	
Seq	
TTL	
C	11
B	7
A	1

C

Seq	
TTL	
D	11
B	3



A	
Seq	
TTL	
B	2
D	1

B	
Seq	
TTL	
A	2
D	7
C	3

Step 2 − Every router flood the link state packets to every offer router

At A −

Link state packet B, C, D

From B

A	2
C	3
D	7

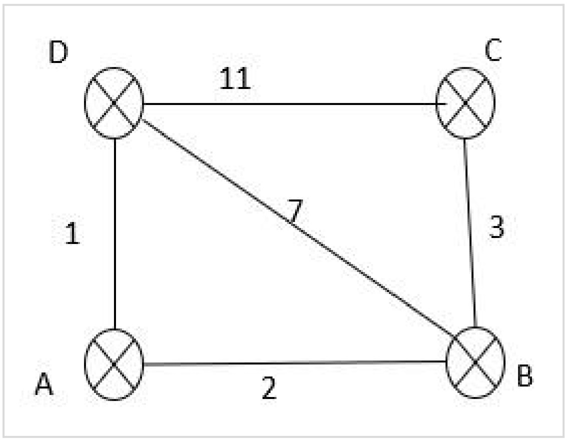
From C

B	3
D	11

From D

A	1
B	7
C	11

Now A can construct the entire graph using the received link protocol.



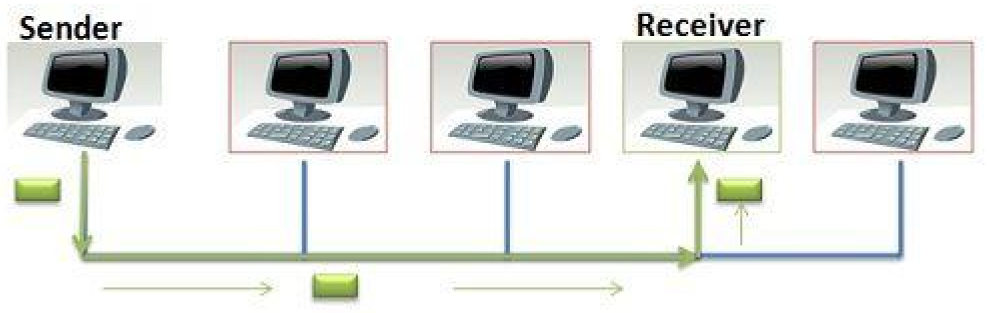
Like this, every node is able to construct the graph in its own memory. Every node has an entire graph. So every router can apply the Dijkstra algorithm to find the shortest path.

Unicast

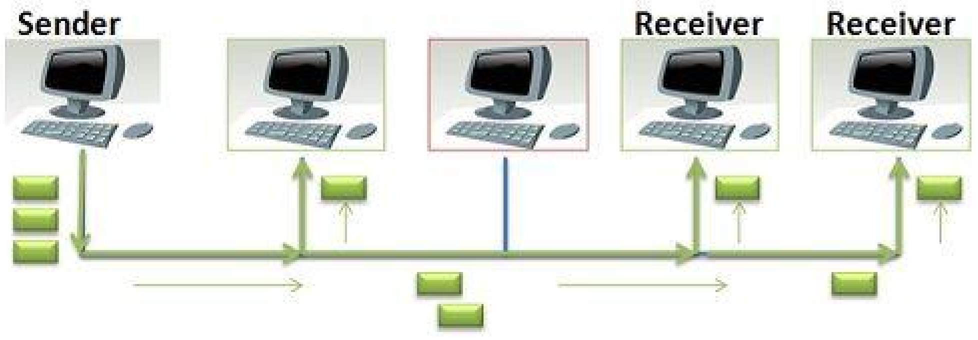
In Computer Networks, the term **unicast** is a transmission method where one station sends information to another station. It is a one-to-one communication. Unicast transmission is used, where one station transmits some private or unique information to another station.

Examples of the unicast transmission are web surfing, file transfer as here the there is a single service requestor and a single service provider.

If one station needs to send packets to multiple stations, it has to send multiple unicast packets, each packet containing the address of the specific station and it is called "**multiple unicasting**". Multiple unicasting utilizes the maximum bandwidth of the network. TCP protocol supports unicasting.



Unicast



Multiple Unicast

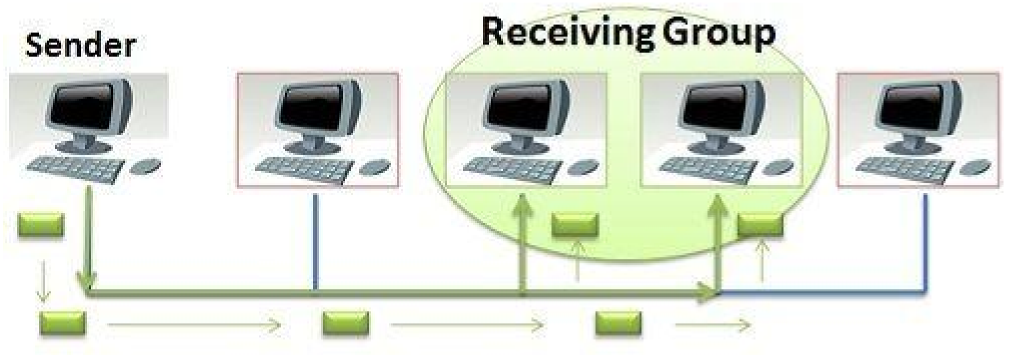
In the above figure, I had shown both the unicasting and multiple unicasting. In unicast clearly shows that the sender is sending the packet to only one receiver station which s highlighted by green color and rest station highlighted by green color are non-receiving stations.

Now look at the figure of multiple unicasting, The sender is needed to send the packet to three receiving stations so, it has created three separate packets, containing the address of three separate receiving stations and each packet is delivered to address on it.

Multicast

Multicast is an information transmission method where one station transmits the information packet to the interested stations only. It is a one-to-many communication method. It is a mixture between unicast and broadcast, where unicasting sends the packet to only one station, and broadcasting sends the packet to all the stations, their multicasting sends the packet to only some selected stations in the network.

Examples of multicasting are forwarding emails, multimedia delivery, etc.



Multicast

In the figure of multicast, you can clearly see that the sender station has created a single packet only which now will be delivered to the group of interested stations only. A single packet is forwarded to the group of receiving stations.

It's hard to use multicasting across a large network because only small sections of the internet are multicast-enabled. Multicast utilizes the bandwidth of the network very efficiently. The group of the receiving stations is decided dynamically. Multicast uses a UDP transport protocol.