**Experiment No.:**

**AIM:** Implement Unicast Routing Algorithm. (Distance Vector Routing Algorithm).

**Objective:** **Students should be** Able to develop a program on Distance Vector Routing Algorithm.

**Theory:**

**Routing Algorithm:**

The routing algorithm is that part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on. If the subnet uses datagrams internally, this decision must be made anew for every arriving data packet since the best route may have changed since last time. If the subnet uses virtual circuits internally, routing decisions are made only when a new virtual circuit is being set up.

Routing algorithms can be grouped into two major classes: nonadaptive and adaptive.

Nonadaptive algorithms do not base their routing decisions on measurements or estimates of the current traffic and topology.

Adaptive algorithms, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well. Adaptive algorithms differ in where they get their information (e.g., locally, from adjacent routers, or from all routers), when they change the routes and what metric is used for optimization (e.g., distance, number of hops, or estimated transit time).

**Distance vector Routing Algorithm :**

Distance vector routing algorithms operate by having each router maintain a table (i.e, a vector) giving the best known distance to each destination and which line to use to get there. These tables are updated by exchanging information with the neighbors. The distance vector routing algorithm is sometimes called by other names, most commonly the distributed Bellman-Ford routing algorithm and the Ford-Fulkerson algorithm.

In distance vector routing, each router maintains a routing table indexed by, and containing one entry for, each router in the subnet. This entry contains two parts: the preferred outgoing line to use for that destination and an estimate of the time or distance to that destination. The metric used might be number of hops, time delay in milliseconds, total number of packets queued along the path, or something similar. The router is assumed to know the ''distance'' to each of its neighbors. If the metric is hops, the distance is just one hop. If the metric is queue length, the router simply examines each queue. If the metric is delay, the router can measure it directly with special ECHO packets that the receiver just timestamps and sends back as fast as it can.

**Example:**

**Draw subnet & solve all steps**

**The Count to Infinity Problem:**

**Refer Lab manual for description & example**

**Program:**

**Output:**

**Conclusion:** The least-cost route between any two nodes is the route with minimum distance. Each node maintains a vector of minimum distances to every node