Department of Computer Science & Engineering , University of Dhaka

CSE-3111, Computer Networking Lab
Assignment 2:
Distance Vector Routing

Submitted by:
Mirajul Mohin, Roll: FH-28

Date of Submission:
30 April 2018

Submitted to:
Upama Kabir, Professor, Dept. of CSE, DU

Design Description:

Distance Vector Routing Protocol (DVRP) is one of two major routing protocols for communications methods that use data packets sent over Internet Protocol (IP). DVRP requires routing hardware to report the distances of various nodes within a network or IP topology in order to determine the best and most efficient routes for data packets. Distance vector routing is a simple routing protocol used in packet-switched networks that utilizes distance to decide the best packet forwarding path. Distance is typically represented by the hop count.
Routing protocols that use distance-vector routing protocols include RIP (Routing Information Protocol), Cisco's IGRP (Internet Gateway Routing Protocol), and Apple's RTMP (Routing Table Maintenance Protocol).

A distance-vector routing (DVR) protocol requires that a router inform its neighbors of topology changes periodically. Historically known as the old ARPANET routing algorithm (or known as Bellman-Ford algorithm).

Bellman Ford Basics:

Each router maintains a Distance Vector table containing the distance between itself and ALL possible destination nodes. Distances,based on a chosen metric, are computed using information from the neighbors' distance vectors.

Information kept by DV router:

- Each router has an ID
- Associated with each link connected to a router,there is a link cost (static or dynamic).
- Intermediate hops

Distance Vector table initialization:

- Distance to itself = 0
- Distance to ALL other routers = infinity number.

Distance Vector Algorithm :

- A router transmits its distance vector to each of its neighbors in a routing packet.
- Each router receives and saves the most recently received distance vector from each of its neighbors.
- A router recalculates its distance vector when:
    - It receives a distance vector from a neighbor containing different information than before.
    - It discovers that a link to a neighbor has gone down.

The DV calculation is based on minimizing the cost to each destination:

Dx(y) = Estimate of least cost from x to y
C(x,v) = Node x knows cost to each neighbor v
Dx = [Dx(y): y ∈ N] = Node x maintains distance vector
Node x also maintains its neighbors' distance vectors
– For each neighbor v, x maintains Dv = [Dv(y): y ∈ N]

Note :

- From time-to-time, each node sends its own distance vector estimate to neighbors.
- When a node x receives new DV estimate from any neighbor v, it saves v's distance vector and it updates its own DV using B-F equation:
  => Dx(y) = min { C(x,v) + Dv(y)} for each node y ∈ N

A simple simulation of Distance vector Routing algorithm is given below:

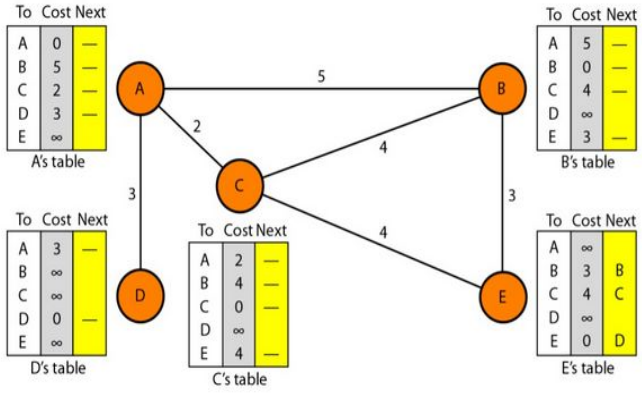

Figure-1: Initial distance vector of 5 nodes

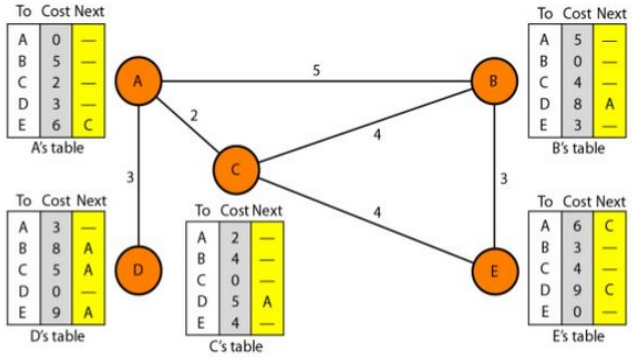

Figure-2: Final distance vector of 5 nodes

The java project contains four classes (dv_routing.java, dv_send.java, dv_Receive.java, Packet_Format.java). dv_routing.java contains main class and has some methods of taking command line arguments and printing the final distance vector. Main method initializes the distance vector of each node (for a node, distance to itself=0 , for others node= infinity). If code is run once, main method will start the dv_Send.java and dv_Receive.java threads. Main thread waits for 3 min to complete the execution of other two threads. Then main thread prints the distance vector of each node.

dv_Send.java thread sends the distance vector of a node to all of its neighbours(line no. 133,134). In the run() method, a node sends its distance vector to all of its neighbours and waits for 5 sec(line no. 143) before sending again.

dv_Receive.java thread receives the distance vector of all adjacent node of a node and checks the condition of updating distance vector. If condition satisfies, distance vector is updated or not (line no. 185,186).

Packet Format:

Packet_Format.java class contains a constructor which forms the packet to send. Packet contains source, destination, next hop, distance, sender IP, Receiver IP. As Sender IP and Receiver IP are same (127.0.0.1), it is only used for making a nice packet format, not for further use. Packet_Format.java class has the packet format(line no 206-213). Packet format used in dv_Routing.java (line no. 65,68).

Poison Reverse is not handled here.

References:

- https://www.geeksforgeeks.org/computer-network-routing-protocols-set-1-distance-vector-routing/
- http://www.linfo.org/distance_vector.html
- http://ecomputernotes.com/computernetworkingnotes/routing/explain-distance-vector-routing-protocol-in-detail