Project 4 : Distance Vector  
CSCI P538 Computer Networks  
Group Member: Shengyu Chen, Xiner Zhang

According to the requirement, we need to implement the distributed asynchronous distance vector.

From the textbook, a distance-vector routing protocol in data networks determines the best route for data packets based on distance. Distance-vector routing protocols measure the distance by the number of routers a packet has to pass, one router counts as one hop. Some distance-vector protocols also take into account network latency and other factors that influence traffic on a given route. To determine the best route across network routers on which a distance-vector protocol is implemented exchange information with one another, usually routing tables plus hop counts for destination networks and possibly other traffic information. Distance-vector routing protocols also require that a router informs its neighbor of network topology changes periodically. Furthermore, distance-vector routing protocols use the Bellman–Ford algorithm and Ford–Fulkerson algorithm to calculate the best route. Another way of calculating the best route across a network is based on link cost, and then is implemented through link-state routing protocols. The term distance vector refers to the fact that the protocol manipulates vectors (arrays) of distances to other nodes in the network. The distance vector algorithm was the original ARPANET routing algorithm and was implemented more widely in local area networks with the Routing Information Protocol (RIP).

For the graph data structure, which contain a list of vertices and a list of edge. After we apply the configuration file in the structure, the initializefunction will set the initial values in the graph, achieved by Bellman-Ford algorithm.

According to the graph provided in the figure 2 in the assignment file, we design the graph table. For each vertex in the graph should have a label that identifies the vertex and a calculated distance for reaching the node from a specified source. The weight for edges (A, B), and (A,C) is 1. The weight for edge (A, D) is Infinity. Note that A’s configuration file does not indicate the relationships between, nodes B and D, B and C or C and D. The Bellman-Ford algorithm would first set the distance for each vertex to Infinity, followed by setting the distance for the source A to 0. Finally, we apply poison reverse and split horizon into routing information protocol(RIP). With split horizon, A and B immediately stop announcing the default route to each other — they get rid of the loop as soon as the route times out; With poison reverse, A and B announce an infinite metric default route to each other, which gets rid of the routing loop as soon as an update is successfully transmitted.