University of Dhaka

Department of Computer Science and Engineering

CSE 3111 – Computer Networking Laboratory Credits: 1.5 Batch: 26/3rd Year 1st Sem 2022

Instructors: Prof. Dr. Md. Abdur Razzaque (AR), Mr. Md. Mahmudur Rahman (MRR), Mr. Md. Ashraful Islam (MAI) and Mr. Md. Fahim Arefin (FA)

Lab Experiment # 8

Name of the Experiment: Implementation of Distance Vector Algorithm

Objective:

In this lab, you will be exploring the Distance Vector Routing Algorithm, which is a dynamic routing algorithm that computes the shortest path to each destination in a network. This algorithm is based on the Bellman-Ford algorithm and is used by distance-vector routing protocols such as RIP (Routing Information Protocol). You will simulate the algorithm's operation in a network topology and observe how routing tables are updated.

Procedure:

- Design a program that simulates a network of routers, where each router is represented as a node, and each link is represented as an edge with a given cost.
- Each router will have a mapping of **port and router names**.
- Implement the Distance Vector Algorithm including the following components:

1. Initialize the Network Topology: Create a data structure (e.g., an adjacency matrix or list) to represent the network topology. Define the nodes (routers) and edges (links) with their corresponding costs or metrics. Set the distance to directly connected networks to the link cost and the distance to non-directly connected networks to infinity.
2. Initialize Routing Tables: Initialize the routing tables for each router in the network. The routing table for each router should contain entries for all other routers in the network, with the initial distances set based on direct connections.
3. Define the Update Function: Define a function that updates the routing table of a router based on the Bellman-Ford equation: new_distance = min(old_distance, neighbor_distance + link_cost). The function should take distance vector updates from neighboring routers and update the routing table accordingly.
4. Exchange Distance Vectors: Simulate the periodic exchange of distance vector updates between neighboring routers. Each router should send its current distance vector (routing table) to its directly connected neighbors.
5. Update Routing Tables: After each exchange, call the update function to recalculate the distances in the routing tables based on the received distance vectors from neighbors. Update the next-hop information for each destination as well.
6. Check for Convergence: Determine whether the network has converged by checking if the routing tables have stabilized (i.e., no further changes occur in the distances). If the network has not converged, repeat the exchange and update steps. Print the gradual updates of path in each node.
7. Analyze the Results: Analyze the final state of the routing tables, verify the correctness of the computed shortest paths, and observe the algorithm's behavior in terms of convergence and response to network changes.
8. Introduce Changes (Optional during Lab): Optionally, you can introduce changes to the network, such as modifying link costs or simulating link failures, to observe how the algorithm responds and reconverges.
9. Test the functionality of the Algorithm in various network scenarios by changing the network topology and evaluating the calculated shortest paths. This can be done by using a 30 second timer which randomly updates an edge [The initial network topology should be configurable from a file]

Results:
Include the following in your lab report:

- A detailed description of the program, including the implementation of the Distance Vector Algorithm, the network topology used.
- Screenshots or console output demonstrating the correct functionality of the program in various network scenarios.
- A discussion of any issues or challenges encountered during the implementation process and the steps taken to resolve them.
- Comparison between Link state and Distance Vector algorithm.

Conclusion:
Summarize the results of the experiment and discuss the key takeaways from implementing the Distance Vector Algorithm.

References:
Include any relevant references or resources used during the lab experiment, such as textbooks, online tutorials, or official documentation.

Sample Input Graph :

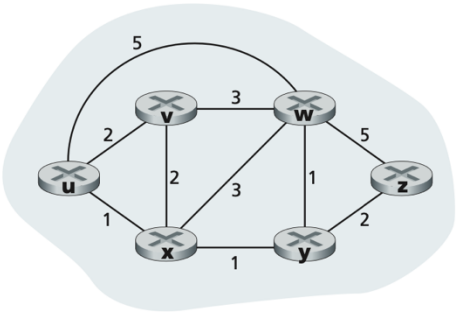

Figure 5.3 ♦ Abstract graph model of a computer network