| Name of the Student: Neha Gupta |
| --- |
| Class and Div and Roll No: TE-1 / 49 |
| Experiment No.10 |
| Implement a routing using the Distance Vector routing  algorithm. |
| Date of Performance: 09/09/2025 |
| Date of Submission: 13/09/2025 |

**Experiment No 10**

**Aim: Implement a routing using the Distance Vector routing algorithm.**

**Objectives:** To configure and analyze routing using the Distance Vector routing algorithm (RIP) in order to understand how routers exchange information and determine optimal paths in a network.

**Theory**: Distance Vector Routing (DVR) is a dynamic routing protocol in which each router maintains a table (vector) containing the minimum distance to reach every other node in the network. These distances are updated by periodically exchanging information with directly connected neighbors.

- Each router knows the cost to reach its immediate neighbors.  
- Routers exchange their distance vectors with neighbors at regular intervals.  
- The Bellman-Ford algorithm is used to update routing tables.  
- If a router learns about a shorter path through a neighbor, it updates its table.  
  
Formula:  
 D\_x(y) = min\_v { C\_xv + D\_v(y) }  
Where:  
- D\_x(y) = Distance from node x to node y  
- C\_xv = Cost from x to neighbor v  
- D\_v(y) = Distance from neighbor v to y  
  
Advantages:  
- Simple to implement.  
- Works well for small and medium-sized networks.  
  
Disadvantages:  
- Slower convergence.  
- Count-to-infinity problem.

## ALGORITHM

1. Start
2. Initialize the cost table with given direct connection costs; set cost to infinity where there is no direct connection.
3. For each router, maintain a distance vector containing the cost to reach every other node.
4. Send the distance vector to all directly connected neighbors.
5. Upon receiving a distance vector from a neighbor:  
    - For each destination:  
    - Calculate the new cost = cost to neighbor + neighbor’s cost to destination.  
    - If the new cost is less than the current stored cost, update the routing table.
6. Repeat steps 4–5 until no updates occur (convergence).
7. Display the final routing table for each router.

Stop

Code:

#include <stdio.h>

#include <stdbool.h>

#define INF 9999

#define N 8 // Number of nodes

void printDistanceMatrix(int distance[N][N]) {

printf("Distance matrix:\n");

for (int i = 0; i < N; i++) {

for (int j = 0; j < N; j++) {

if (distance[i][j] == INF)

printf("INF ");

else

printf("%4d ", distance[i][j]);

}

printf("\n");

}

}

void distanceVectorRouting(int cost[N][N]) {

int distance[N][N];

// Initialize distance matrix with cost matrix

for (int i = 0; i < N; i++) {

for (int j = 0; j < N; j++) {

distance[i][j] = cost[i][j];

}

}

bool updated;

do {

updated = false;

for (int i = 0; i < N; i++) {

for (int j = 0; j < N; j++) {

for (int k = 0; k < N; k++) {

if (distance[i][k] + distance[k][j] < distance[i][j]) {

distance[i][j] = distance[i][k] + distance[k][j];

updated = true;

}

}

}

}

} while (updated);

printDistanceMatrix(distance);

}

int main() {

int cost[N][N] = {

// 0 1 2 3 4 5 6 7

{ 0, 3, INF, INF, 7, INF, INF, INF}, // 0

{ 3, 0, 4, INF, 2, 6, INF, INF}, // 1

{INF, 4, 0, 5, INF, 1, INF, INF}, // 2

{INF, INF, 5, 0, INF, INF, 3, 8}, // 3

{ 7, 2, INF, INF, 0, 3, INF, INF}, // 4

{INF, 6, 1, INF, 3, 0, 4, INF}, // 5

{INF, INF, INF, 3, INF, 4, 0, 2}, // 6

{INF, INF, INF, 8, INF, INF, 2, 0} // 7

};

printf("Initial Cost Matrix:\n");

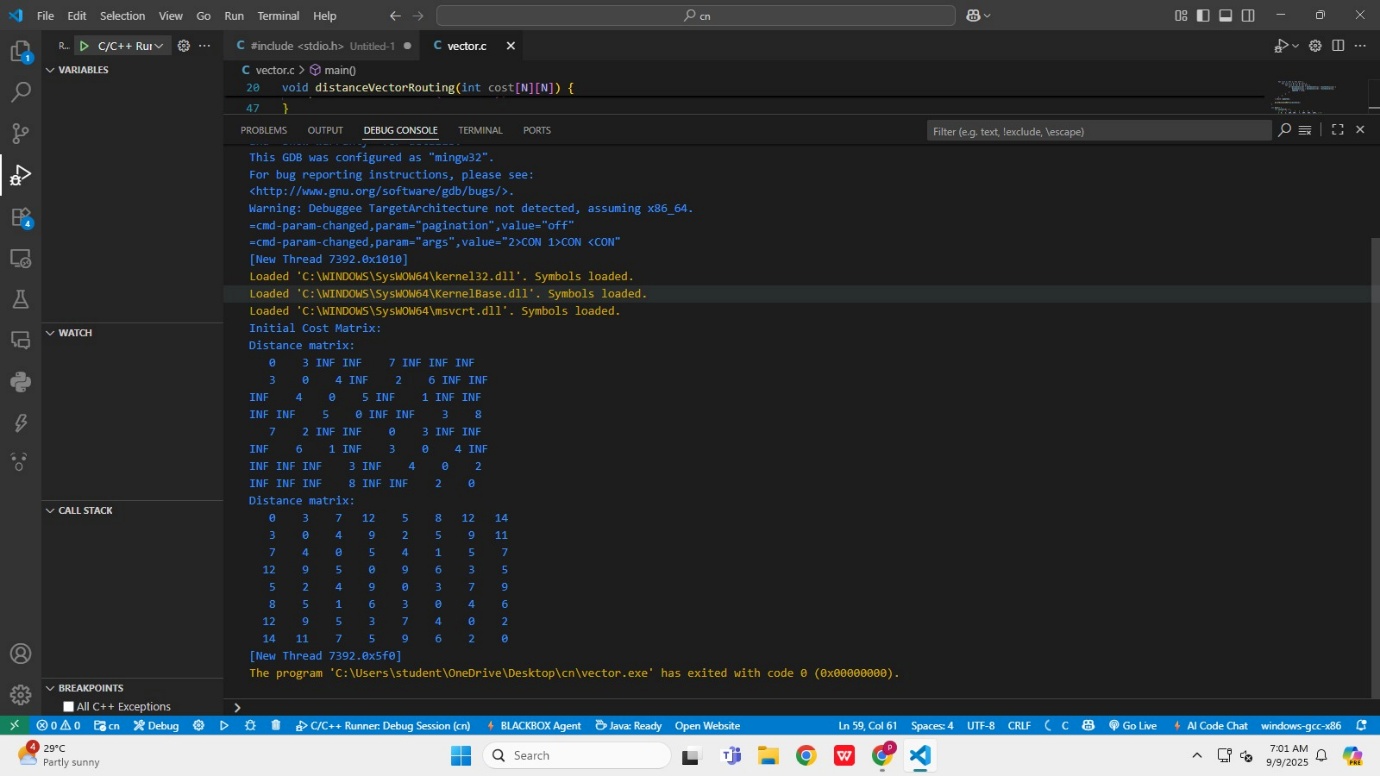
printDistanceMatrix(cost);

distanceVectorRouting(cost);

return 0;

}

**Output:**



**Conclusion:** Write your understanding of how the Distance Vector Routing algorithm works, what you have observed in the output, and how the routing table converged

**Questions:**

1. Why is RIP not suitable for large networks?

Ans-

* RIP has a maximum hop count of 15, so it cannot support large networks with many routers.
* It converges slowly, causing delays if the network changes.
* RIP only uses hop count as a metric, ignoring link speed and reliability, which can lead to inefficient routing paths.
* The protocol creates unnecessary network traffic by periodically broadcasting updates, increasing overhead.
* It lacks modern security features and advanced capabilities needed in big, complex networks.

1. How can triggered updates improve RIP performance?

Ans-

* Triggered updates allow RIP to send immediate routing updates when a topology change occurs, instead of waiting for the next scheduled update.
* This mechanism speeds up convergence, so all routers learn about changes (like a failed route) much faster.
* Faster updates help reduce routing loops and network downtime, improving overall RIP performance.
* Only the affected network information is sent, saving bandwidth compared to full periodic updates.

1. If two routes have the same hop count in RIP, how does the router decide the path?

Ans-

* If two routes have the same hop count in RIP, the router usually performs load balancing and can use both routes to forward packets.
* RIP supports equal-cost load balancing, so the router sends packets over all available paths with the same metric.
* If load balancing is not enabled, the router may just select the first route it learns or the route with the lowest next-hop IP address, depending on implementation.

1. Differentiate between Distance Vector and Link State routing algorithms with examples.

Ans-

|  |  |  |
| --- | --- | --- |
| Parameter | Distance Vector algorithm | Link State routing algorithm |
| Routing Information  Exchange | Routers share their routing tables with immediate neighbors only  (e.g., RIP). | Link State: Routers flood the network with information about their own links, giving all routers a global view (e.g., OSPF). |
| Algorithm Used | Uses the Bellman-Ford algorithm. | Uses Dijkstra’s algorithm. |
| Knowledge Scope: | Routers know only distance (cost) to destinations via neighbors | Link State: Routers know the full network topology |
| Parameter | Distance Vector algorithm | Link State routing algorithm |
| Convergence | Slower convergence; prone to routing loops and count to infinity problems. | Faster convergence; fewer routing loops. |
| Bandwidth and Resources | Lower resource and bandwidth usage (periodic table updates). | Higher resource and bandwidth usage (link state advertisements and database maintenance). |
| Examples | RIP (Routing Information  Protocol) | OSPF (Open Shortest Path First), IS-IS |

1. Can RIP be used in modern enterprise networks? Justify your answer.

Ans- RIP is generally not suitable for modern enterprise networks for several reasons:

* Limited scalability: RIP’s maximum hop count is 15, limiting network size and complexity.
* Slow convergence: It takes longer to update routes after topology changes, which can cause network instability.
* Inefficient routing: RIP uses only hop count as a metric, ignoring important factors like bandwidth and delay, leading to suboptimal routing.
* High bandwidth consumption: Periodic broadcasting of routing tables regardless of changes wastes network resources.
* Lack of security features: RIP lacks modern security mechanisms, making it vulnerable in today's environments.

Due to these limitations, modern enterprises typically choose more advanced routing protocols like OSPF or EIGRP that support faster convergence, better metrics, scalability, and security features. Therefore, RIP is not recommended for contemporary enterprise networks