**深 圳 大 学 实 验 报 告**

**课程名称：­ 计算机网络（Computer Networks）**

**实验名称： Network Layer Assignment**

**学院： 电子与信息工程学院**

**专业： 电子信息工程**

**指导教师： 毕宿志**

**报告人： 陈应权 班级： 06 学号： 2022280297**

**实验时间： 2023年12月1日--2023年12月15日**

**实验报告提交时间： 2023年12月15日**

**教务部制**

|  |
| --- |
| 1. **Purpose of experiment**   **①Implement the Dijkstra algorithm in Python to find the shortest path and distance from a specified node to all other nodes in a network.**  **②Develop a Distance-Vector algorithm in Python for determining the shortest path and distance from a specified node to all other nodes in a network.**  **③Through this experiment, deepen our understanding of Dijkstra algorithm and Distance-Vector algorithm .Besides, we are able to clearly understand the difference between the two algorithms**   1. **Experimental principle**   **①Dijkstra algorithm**  **Dijkstra's algorithm is a popular algorithm used to find the shortest path from a single source vertex to all other vertices in a weighted graph. It operates by iteratively exploring nodes, starting from the source node and gradually determining the shortest paths to all other nodes.**  **②Distance-Vector algorithm**  **The Distance-Vector algorithm, also known as the Bellman-Ford algorithm, is a distributed routing algorithm used in computer networks to calculate the shortest path between nodes. Unlike Dijkstra's algorithm, which is a centralized algorithm, Distance-Vector operates in a decentralized manner, where each node iteratively exchanges routing information with its neighboring nodes to compute the shortest paths.**   1. **Content**   **(一)Dijkstra algorithm**  **①Initialization**  **Assign a tentative distance value to every node in the graph. Set the distance to the source node as 0 and all other nodes' distances as infinity.**  **Create an empty priority queue (min-heap or similar) to store nodes based on their tentative distances.**  **②Iterative Exploration:**  **While there are nodes to visit, repeatedly select the node with the smallest tentative distance from the priority queue.**  **③Node Evaluation:**  **For the selected node (start\_node), examine all of its neighboring nodes (adjacent nodes) and update their tentative distances if a shorter path is found.**  **Calculate the tentative distance from the source node to each neighboring node via the start\_node. If this distance is shorter than the previously recorded distance, update it.**    **④Mark Visited Nodes:**  **After evaluating all the neighbors of the start\_node, mark it as visited to avoid redundant evaluations.**  **⑤Repeat Until Completion:**  **Continue this process, selecting the node with the smallest tentative distance as the start\_node until all nodes have been visited or until the destination node (if specified) is reached.**  **⑥Output:**  **The algorithm concludes when all reachable nodes have been visited, and the shortest path and distances from the source node to all other nodes are determined.**  **(二) Distance-Vector algorithm**  **①Initialization:**  **Each node maintains a routing table that includes the distance (cost) to reach all other nodes in the network and the next-hop information to reach those nodes.**  **②Distance Calculation:**  **Nodes exchange their routing tables with their immediate neighbors.**  **Upon receiving a routing table from a neighbor, a node updates its own routing table based on the information received. It recalculates the distance to each destination by considering the minimum cost path through its neighbors.**  **③Iterative Update:**  **Nodes continuously exchange their routing tables with neighbors and update their own tables based on the received information.**  **Each iteration involves recalculating distances and updating routing tables until no further changes occur in the tables or until convergence is reached.**  **④Distance Vector Updates:**  **When a node's routing table is updated due to information received from a neighbor, it propagates this change to its neighboring nodes by broadcasting updated distance vectors.**  **⑤Convergence:**  **The algorithm continues until all nodes' routing tables converge, meaning that no further changes occur in the tables. At this point, each node has accurate routing information.**  **⑥Output:**  **The algorithm concludes when all reachable nodes have been visited, and the shortest path and distances from the source node to all other nodes are determined.** |
| 1. **Conclusion and discussion**   **Through this experiment, I deepened my understanding of the two algorithms and also understood the differences between the two algorithms.**  **①Dijkstra Algorithm:**  **Dijkstra's algorithm, being a centralized approach, demonstrated its effectiveness in determining the shortest path from a specified source node to all other nodes in a network. Its reliance on maintaining a priority queue for node selection ensures accuracy in path determination. However, its computational complexity can be a concern for larger networks, and its suitability may vary based on the specific application and network characteristics.**  **②Distance-Vector Algorithm:**  **The Distance-Vector algorithm, on the other hand, showcased a decentralized approach to routing. By iteratively exchanging routing tables between neighboring nodes, it adapts to changes in the network topology. This flexibility is valuable in dynamic network environments. However, the algorithm may experience slower convergence and has the potential for issues such as the "count-to-infinity" problem, where inaccurate information can persist during updates.** |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日  备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。