Progetto Python: Distance Vector Routing

Jacopo Turchi

12 dicembre 2024

Indice

1 Introduzione **2**

2 Descrizione dei File **3**
2.1 `main.py` 3
2.2 `network_setup.py` 3
2.3 `router_management.py` 3

3 Funzionamento dello Script **5**
3.1 Creazione della Rete Iniziale 5
3.2 Modifica della Rete 7
3.3 Ulteriore Modifica della Rete 10

4 Conclusione **13**

Capitolo 1

Introduzione

Ho scelto di seguire la seconda traccia, realizzando uno script Python che simula il protocollo *Distance Vector Routing*. Lo script è composto da tre file principali: `main.py`, `network_setup.py`, e `router_management.py`.

Capitolo 2

Descrizione dei File

2.1 `main.py`

Il file `main.py` è il punto di ingresso dello script. Esso crea una rete iniziale utilizzando le funzioni definite in `network_setup.py` e simula il protocollo di Distance Vector Routing utilizzando le funzioni definite in `router_management.py`. Inoltre, modifica dinamicamente la rete e testa il corretto funzionamento in diverse condizioni.

2.2 `network_setup.py`

Il file `network_setup.py` contiene funzioni per la creazione e la modifica della rete. Le funzioni principali includono:

- `add_node(network, node)`: Aggiunge un nuovo nodo alla rete.
- `add_edge(network, node1, node2, cost)`: Aggiunge un collegamento tra due nodi con un costo specificato.
- `remove_edge(network, node1, node2)`: Rimuove il collegamento tra due nodi.
- `remove_node(network, node)`: Rimuove un nodo e i suoi collegamenti dalla rete.
- `create_network()`: Crea una rete iniziale di partenza aggiungendo nodi e collegamenti con costi specificati.

2.3 `router_management.py`

Il file `router_management.py` contiene funzioni per la gestione delle tabelle di routing e la simulazione del protocollo Distance Vector. Le funzioni principali includono:

- `initialize_routing_table(network)`: Inizializza le tabelle di routing per ogni nodo nella rete.
- `update_routing_table(node, neighbors, routing_table)`: Aggiorna la tabella di routing di un nodo usando l'algoritmo di Bellman-Ford.
- `propagate_updates(network, routing_table)`: Propaga gli aggiornamenti delle tabelle di routing fino alla convergenza.
- `print_routing_tables(routing_table)`: Stampa le tabelle di routing in formato tabellare.
- `simulate_distance_vector_routing(network)`: Esegue la simulazione del Distance Vector Routing.

Capitolo 3

Funzionamento dello Script

3.1 Creazione della Rete Iniziale

Lo script inizia creando una rete iniziale con tre nodi (`A`, `B`, `C`) e collegamenti con costi specificati. La rete iniziale è rappresentata nella Figura 3.1.

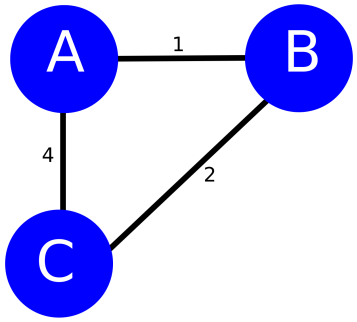

Figura 3.1: Rete iniziale con nodi A, B, C.

Successivamente, esegue la simulazione del Distance Vector Routing, che consiste nell'inizializzare le tabelle di routing e propagare gli aggiornamenti fino alla convergenza La tabella di routing iniziale è mostrata nella Figura 3.2. La tabella finale è mostrata nella Figura 3.3

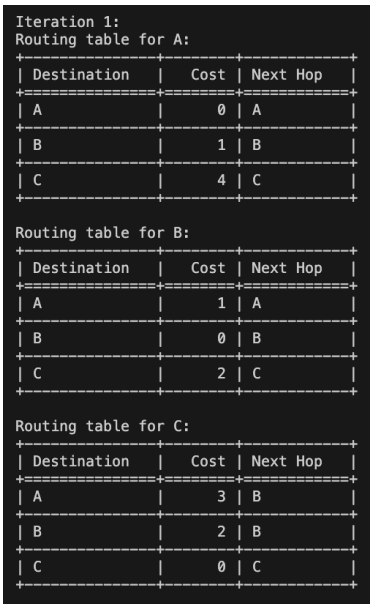

Figura 3.2: Tabella di routing iniziale per i nodi A, B, C.

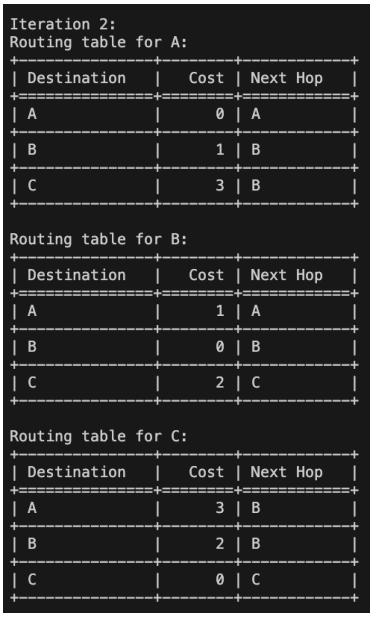

Figura 3.3: Tabella di routing finale per i nodi A, B, C.

3.2 Modifica della Rete

Dopo la prima simulazione, lo script modifica dinamicamente la rete aggiungendo un nuovo nodo (`D`), collegandolo ai nodi esistenti con costi specificati, e rimuovendo il collegamento tra B e C. La rete modificata è mostrata nella Figura 3.4.

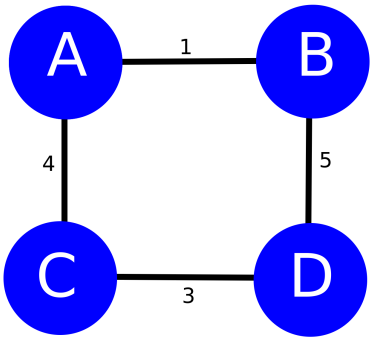

Figura 3.4: Rete modificata con l'aggiunta del nodo D e la rimozione del collegamento tra B e C.

La simulazione viene quindi rieseguita per riflettere i cambiamenti. La tabella di routing iniziale è mostrata nella Figura 3.5. La tabella finale è mostrata nella Figura 3.6

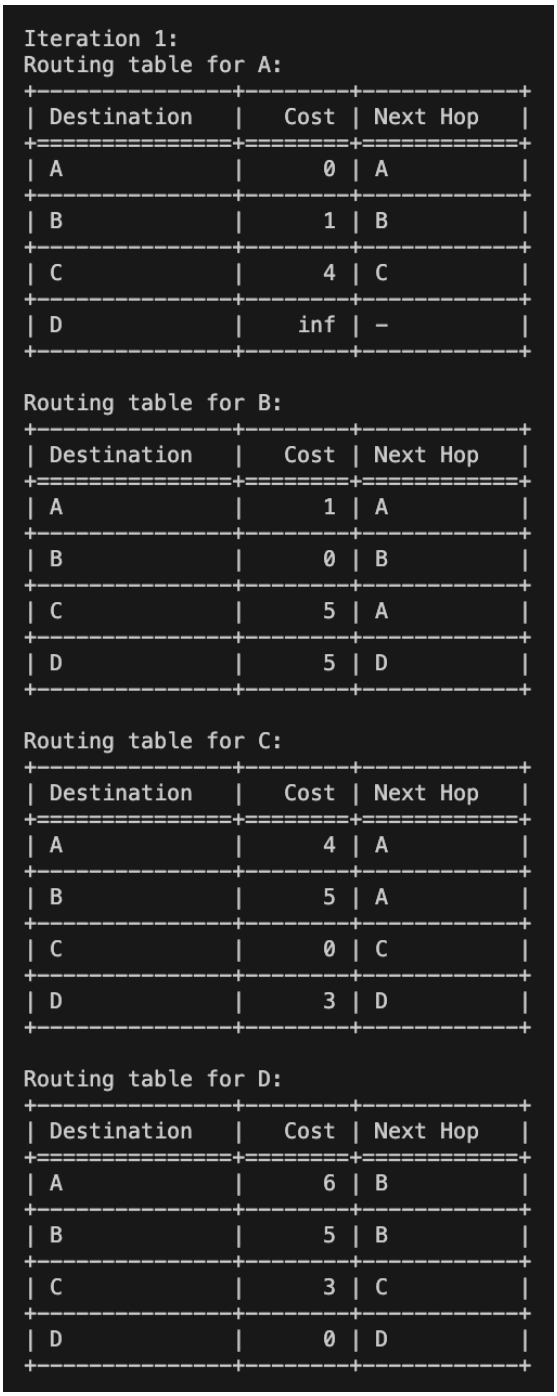

Figura 3.5: Tabella di routing iniziale per i nodi A, B, C, D.

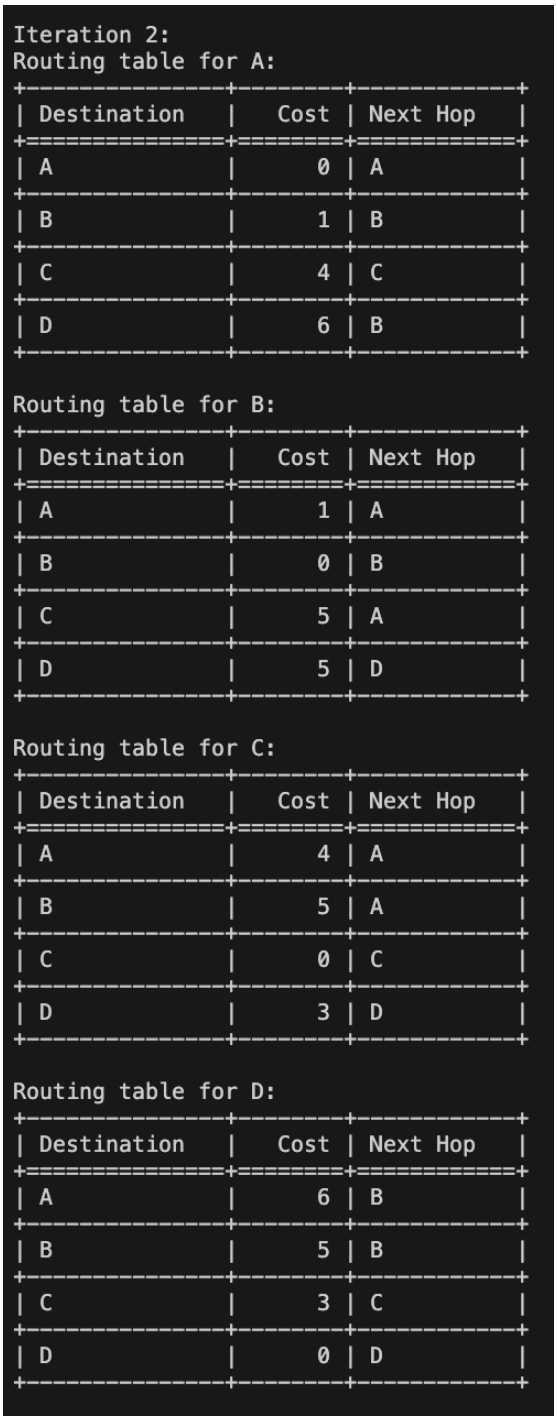

Figura 3.6: Tabella di routing finale per i nodi A, B, C, D.

3.3 Ulteriore Modifica della Rete

Infine, lo script modifica il costo di un collegamento esistente e riesegue la simulazione per aggiornare le tabelle di routing. La rete con il costo modificato è mostrata nella Figura 3.7.

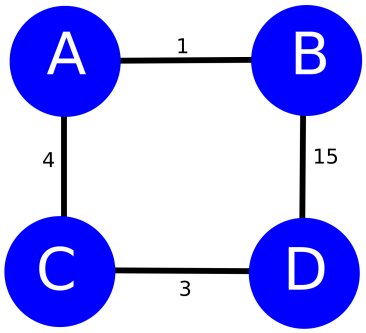

Figura 3.7: Rete con il costo del collegamento tra B e D modificato.

La simulazione viene quindi rieseguita per riflettere i cambiamenti causati dalla modifica del costo dell'arco. La tabella di routing iniziale è mostrata nella Figura 3.8. La tabella finale è mostrata nella Figura 3.9

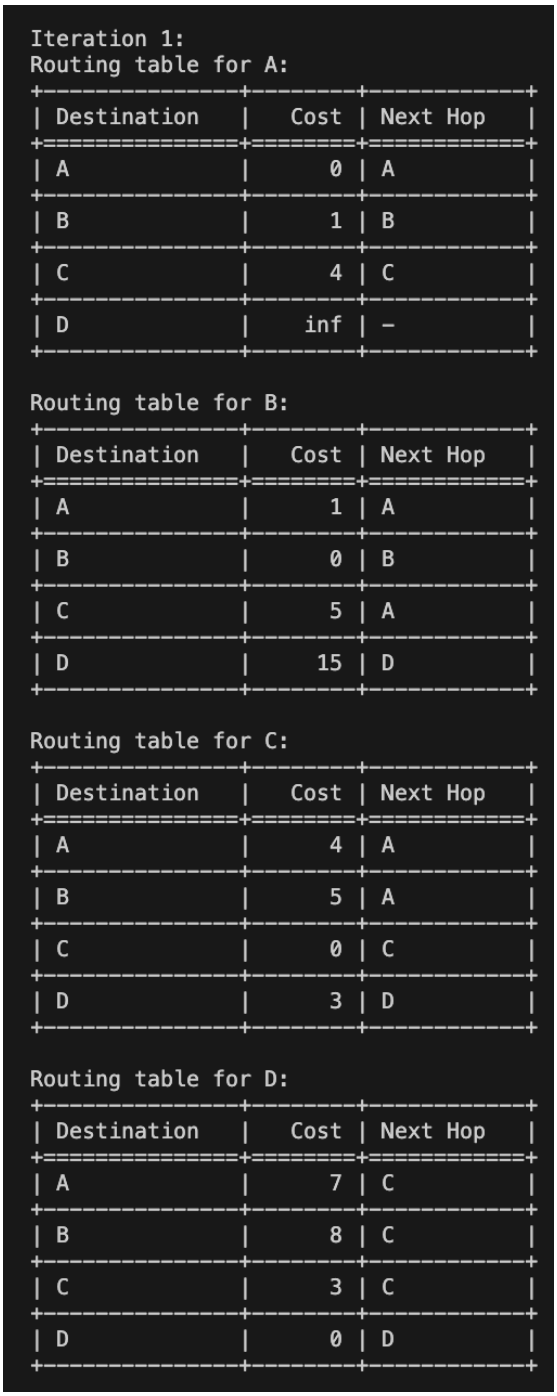

Figura 3.8: Tabella di routing iniziale per i nodi A, B, C, D dopo il cambio di costo.

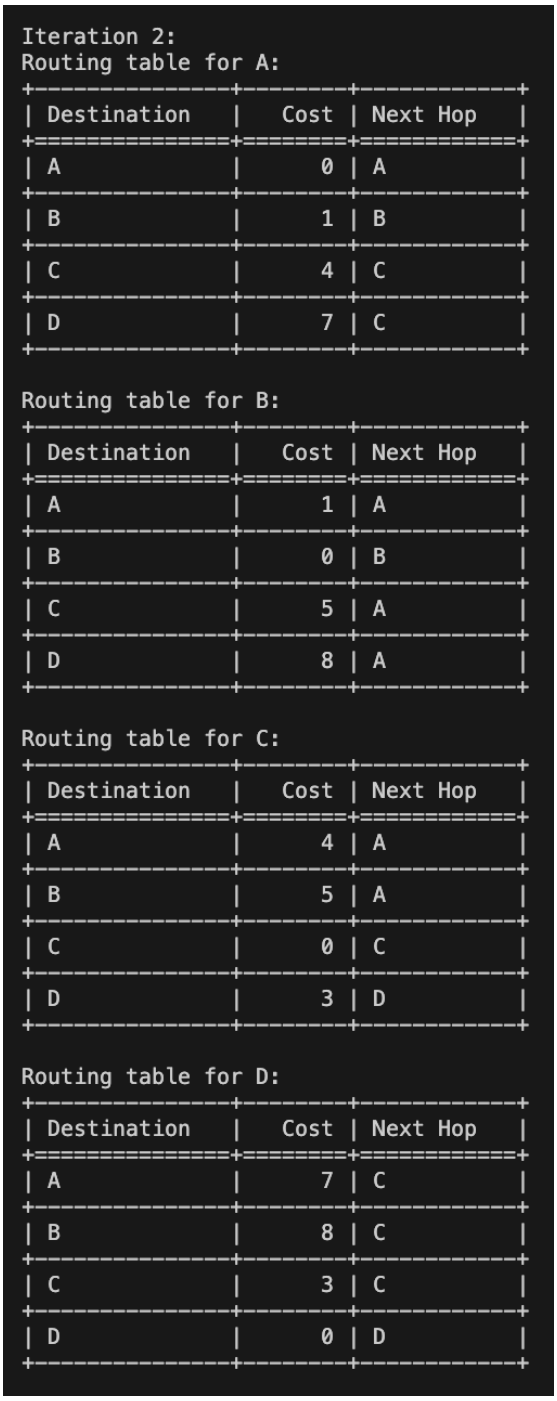

Figura 3.9: Tabella di routing finale per i nodi A, B, C, D dopo il cambio di costo.

Capitolo 4

Conclusione

Questo script fornisce una semplice ma efficace simulazione del protocollo di routing Distance Vector, permettendo di osservare come le tabelle di routing vengono aggiornate in risposta ai cambiamenti nella topologia della rete. Le funzioni modulari e ben definite rendono lo script facilmente estendibile per ulteriori esperimenti e simulazioni.