

Progetto Reti

Distance Vector Routing

Francesco Marcatelli

Dicembre 2024

A.A 2024/2025

Analisi del codice

La soluzione implementa un algoritmo di **routing distribuito basato sulla distanza** (Distance Vector Routing), coprendo gli obiettivi principali:

1. Definizione della rete:

- La classe `Node` rappresenta un nodo con una tabella di routing e una lista dei vicini.
- La classe `Network` rappresenta un'insieme di nodi connessi da link con costi specifici.

2. Inizializzazione delle tabelle di routing:

- La funzione `initialize_table` nei nodi crea le tabelle di routing iniziali per ogni nodo, basandosi sui costi dei link diretti.

```
def initialize_table(self):
    # Inizializza la tabella di routing
    for neighbor, cost in self.neighbors.items():
        self.routing_table[neighbor] = (cost, neighbor)
    self.routing_table[self.name] = (0, self.name) # Nodo a se stesso
```

3. Simulazione del protocollo di routing:

- Il metodo `simulate_routing` itera finché la rete non converge (ossia, finché le tabelle di routing dei nodi non si aggiornano più).

```
def simulate_routing(self):
    converged = False
    iteration = 0
    while not converged:
        converged = True
        iteration += 1
        print(f"\n--- Iteration {iteration} ---")
        for node in self.nodes.values():
            for neighbor in node.neighbors:
                updated = node.update_table(neighbor, self.nodes[neighbor].routing_table)
                if updated:
                    converged = False
        for node in self.nodes.values():
            print(node)
```

4. Output richiesti:

- Ad ogni iterazione, lo stato aggiornato delle tabelle di routing viene stampato, fornendo una chiara rappresentazione di come i percorsi vengono calcolati e aggiornati.

Errori Potenziali e Soluzioni

1. **Non convergenza:**
 - La condizione di convergenza è correttamente implementata usando il flag `converged`. Se nessuna tabella cambia durante un'iterazione, la simulazione si arresta.
2. **Problemi nella topologia della rete:**
 - L'algoritmo è robusto per gestire topologie non connesse. Ad esempio, se un nodo non è raggiungibile, la distanza rimane `inf`.
3. **Diagnosi degli aggiornamenti:**
 - Ogni aggiornamento delle tabelle è chiaramente visibile grazie alla stampa a video ad ogni iterazione.