Relazione: Simulazione del Protocollo di Routing Distance Vector

Introduzione

Il progetto consiste nella simulazione di un protocollo di routing basato sull'algoritmo **Distance Vector Routing**. L'obiettivo principale è quello di modellare un sistema di nodi connessi e permettere a ciascun nodo di aggiornare la propria tabella di routing in base alle informazioni ricevute dai nodi vicini.

Struttura del Progetto

1. File principali

Il progetto è composto da due file:

- `main.py`: Contiene la logica principale della simulazione, incluso l'avvio e la gestione delle iterazioni per l'aggiornamento delle tabelle di routing.
- `routing_table.py`: Definisce la classe `RoutingTable`, che implementa la logica delle tabelle di routing e degli aggiornamenti.

2. Rappresentazione della rete

La rete è modellata come un grafo con archi pesati bidirezionali.

- **Nodi**: Identificati da interi da `0` a `N-1`.
- **Archi**: Definiti da una lista di tuple `(s, e, cost)` che rappresentano i collegamenti tra i nodi `s` ed `e` con un costo associato.

Funzionamento del Codice

1. Inizializzazione

Nel costruttore della classe `RoutingTable`, viene creata una tabella di routing per ciascun nodo, inizialmente vengono inizializzati sono i nodi connessi direttamente e i costi associati.

2. Algoritmo di Aggiornamento

L'algoritmo di aggiornamento segue i seguenti passi:

1. Ogni nodo analizza le distanze dai suoi vicini.
2. Calcola il costo di raggiungere una destinazione attraverso ciascun vicino.
3. Se trova un percorso più corto, aggiorna la propria tabella di routing.

Metodo `update_table(node_id)`

Aggiorna la tabella di routing per un nodo specifico.

Metodo `update()`

Esegue l'aggiornamento delle tabelle di tutti i nodi finche non si verifica la convergenza.

3. Simulazione

La funzione `simulate_routing` nel file `main.py` gestisce la simulazione:

1. Inizializza la rete e le tabelle di routing.
2. Visualizza lo stato iniziale delle tabelle.
3. Itera finché non si verifica la convergenza:
    - Aggiorna le tabelle di routing.
    - Mostra lo stato aggiornato dopo ogni iterazione.

Conclusioni

Il progetto ha dimostrato come i nodi di una rete possano calcolare iterativamente le rotte ottimali utilizzando il protocollo Distance Vector Routing.

Possibili Estensioni

- Supporto per nodi che si aggiungono o rimuovono dinamicamente dalla rete.
- Visualizzazione grafica della rete e delle tabelle di routing.
- Implementazione di algoritmi alternativi come il Link State Routing.