

Relazione sul Protocollo di Routing Simulato

Introduzione

Il presente documento descrive la simulazione di un protocollo di routing basato sull'algoritmo **Distance Vector Routing** implementato in Python. L'obiettivo è dimostrare il funzionamento dell'aggiornamento delle tabelle di routing, il calcolo delle rotte più brevi tra i nodi di una rete e il comportamento del protocollo fino alla convergenza.

Completezza della Soluzione

La simulazione soddisfa tutti gli obiettivi specificati nella traccia:

1. **Configurazione della Rete:** La rete è composta da quattro nodi (**A, B, C, D**) con collegamenti diretti e costi definiti. Ogni nodo gestisce una propria tabella di routing.
2. **Output delle Tabelle di Routing:** Le tabelle vengono stampate a ogni iterazione, mostrando l'evoluzione delle rotte.
3. **Calcolo delle Rotte Più Brevi:** L'algoritmo determina iterativamente il percorso minimo utilizzando l'algoritmo di Bellman-Ford.
4. **Convergenza:** La simulazione termina quando le tabelle smettono di aggiornarsi, dimostrando la corretta propagazione delle informazioni.

L'output include i dettagli richiesti per verificare il comportamento del protocollo.

Chiarezza della Documentazione

Il codice è stato progettato e documentato per garantire la massima leggibilità:

- **Struttura del Codice:** Ogni classe e funzione ha un nome descrittivo, e il codice è organizzato in sezioni logiche.
- **Commenti:** I commenti spiegano i principali blocchi di codice, incluso il funzionamento delle tabelle di routing e il processo di aggiornamento.
- **Output Dettagliato:** L'output delle tabelle di routing è chiaro e include informazioni su ogni destinazione e il costo associato.
- **Relazione Interpretativa:** La relazione descrive il comportamento del protocollo in modo dettagliato, interpretando i dati generati dal codice.

Accuratezza Tecnica

La configurazione e l'implementazione sono coerenti con i principi teorici del Distance Vector Routing:

- **Inizializzazione:** Ogni nodo parte con la conoscenza solo dei collegamenti diretti.
- **Algoritmo Bellman-Ford:** Il calcolo delle rotte si basa sull'aggiornamento iterativo delle tabelle di routing utilizzando i dati ricevuti dai vicini.
- **Convergenza:** La simulazione mostra come le tabelle di routing convergono, confermando che ogni nodo ha identificato i percorsi più brevi.

- **Risultati Attesi:** I percorsi ottimali e i costi finali sono consistenti con la topologia della rete configurata.

Capacità di Diagnosi e Problem Solving

Durante lo sviluppo, sono stati affrontati i seguenti problemi e soluzioni:

1. **Propagazione delle Informazioni:** Inizialmente, alcuni nodi non aggiornavano correttamente le tabelle di routing. Questo è stato risolto verificando che ogni nodo aggiornasse iterativamente le informazioni per tutti i vicini.
2. **Convergenza Prematura:** La simulazione terminava troppo presto. L'algoritmo è stato modificato per ripetere gli aggiornamenti fino a quando nessuna tabella subisce variazioni.
3. **Gestione dei Cicli:** Per evitare problemi legati a percorsi ciclici, è stata aggiunta una logica che previene l'aggiornamento di percorsi che aumentano il costo.

Conclusioni

La simulazione ha dimostrato:

- La corretta implementazione dell'algoritmo Distance Vector Routing.
- La propagazione delle informazioni e la convergenza delle tabelle di routing.
- La robustezza della soluzione nel gestire problemi tecnici e garantire la coerenza dei risultati.

Il progetto è completo, ben documentato e tecnicamente accurato. Potenziali miglioramenti includono l'estensione del modello per gestire dinamicamente variazioni nei collegamenti e implementare ottimizzazioni per ridurre il tempo di convergenza.