Progetto Laboratorio Reti di Telecomunicazioni

A.A 2024/2025

Traccia n.2: Simulazione Protocollo di Routing Distance Vector usando uno script Python

LUCAS ANTONIO LEONTE

0001071582

Obiettivo Generale

Implementare e analizzare un simulatore del protocollo Distance Vector Routing (DVR), con output testuali e visuali, in modo da comprendere in profondità il comportamento iterativo della propagazione delle rotte e della convergenza del sistema. Il progetto mira a verificare:

- La correttezza dell'implementazione logica del protocollo.
- La coerenza tra tabelle di routing e grafico visuale.
- L'allineamento con i principi teorici del Distance Vector Routing.

Requisiti del sistema simulato

- Linguaggio: Python 3.11+
- Librerie: networkx, matplotlib
- Topologia definita: 4 nodi (A, B, C, D) con collegamenti e costi configurabili.
- Output richiesto:
    - Tabelle di routing per ogni nodo ad ogni iterazione.
    - Visualizzazione grafica della rete con distance vector attuali.
    - Rilevamento automatico della convergenza.

Logica implementativa e scelte progettuali

Classe Nodo

Rappresenta un router astratto che contiene:

- La mappa dei vicini con relativi costi.
- La tabella di routing destinazione -> (next_hop, distanza).

Funzioni chiave:

- **inizializza_tabella**: setta le rotte iniziali (zero verso sé, diretto verso i vicini, infinito verso gli altri).
- **invia_vettore**: costruisce il Distance Vector corrente da inviare ai vicini.
- **aggiorna_tabella**: riceve un vettore da un vicino e aggiorna le rotte se trova un percorso più corto.

- **stampa_tabella**: formato leggibile della tabella.

Visualizzazione (**disegna_grafo**)

- Viene aggiornata a ogni iterazione.
- Visualizza graficamente:
    - Nodi e connessioni con costi.
    - Tabella semplificata sotto ogni nodo.
    - Stato corrente della rete.

Simulazione (**simula**)

- Per ogni iterazione:
    - I nodi inviano i propri vettori ai vicini.
    - I vicini aggiornano le proprie tabelle.
    - Si stampa la situazione aggiornata.
    - Se non ci sono aggiornamenti, si dichiara convergenza.

Validazione della correttezza

1. Coerenza tabelle e distanza attesa

Ogni nodo deve calcolare la distanza minima verso ogni altro nodo. Dopo la convergenza, le tabelle devono riflettere i percorsi più brevi, verificabili anche "a mano" tramite l'analisi della topologia.

2. Coerenza visiva

L'output visuale è sincronizzato ad ogni iterazione e riflette le informazioni delle tabelle in tempo reale. Le entry visualizzate sotto ciascun nodo seguono il formato:

Destinazione:NextHop,Distanza

Confrontabili con l'output della stampa_tabella().

3. Comportamento conforme al modello teorico

La rete converge solo quando nessuna tabella viene più aggiornata. Il sistema smette di propagare informazioni inutili dopo la convergenza.

Ragionamento dietro scelte progettuali

- Uso di copy.deepcopy: evita che i nodi si aggiornino tra loro nella stessa iterazione (problema spesso riscontrato nei simulatori concorrenti).
- Visualizzazione iterativa e sincrona: aiuta a comprendere passo passo come si propagano le informazioni e migliorano i percorsi.
- Output testuale + visuale: favorisce il confronto, facilita il debug e permette la validazione incrociata.
- Layout fisso (spring_layout(seed=42)): garantisce stabilità visiva tra le iterazioni.

Conclusioni

Questo simulatore è stato sviluppato con l'obiettivo di facilitare la comprensione del Distance Vector Routing, verificandone passo passo il comportamento, analizzando l'evoluzione delle tabelle e confrontando output visivo e testuale.