Università degli Studi di Bologna
Campus di Cesena

Corso di Laurea Triennale in Ingegneria e Scienze Informatiche

Relazione per il corso di Reti di Telecomunicazioni

Progetto di simulazione routing con Distance Vector

Candidato:
Nicola Graziotin nicola.graziotin@studenti.unibo.it
Matricola 0001081557

Anno Accademico 2024/2025

Indice

1 Classe Main

La classe `Main` gestisce il punto di ingresso del programma, creando i nodi, configurando la rete e avviando la simulazione.

- **Attributi**:
  - `routing`: Istanza della classe `Routing`.
- **Metodi**:
  - `start()`: Esegue i seguenti passi:
    - Crea i nodi e li collega tramite connessioni e distanze.
    - Disegna il grafo della rete tramite `Routing.draw_network()`.
    - Avvia la simulazione di routing con `Routing.routing_simulation()`.

2 Classe Node

La classe `Node` rappresenta un nodo nella rete, dotato di una tabella di routing e di connessioni con i vicini.

- **Attributi**:
  - `name`: Nome del nodo.
  - `routing_table`: Tabella di routing che associa ogni destinazione a una tupla `(distanza, prossimo nodo)`.
  - `neighbors`: Lista dei nodi vicini e delle relative distanze.
- **Metodi**:
  - `add_neighbor(neighbor, distance)`: Aggiunge un nodo vicino e aggiorna la tabella di routing con il percorso diretto.
  - `update_routing()`: Aggiorna la tabella di routing basandosi sulle informazioni dei vicini.
  - `print_routing_table()`: Stampa la tabella di routing del nodo in formato leggibile.

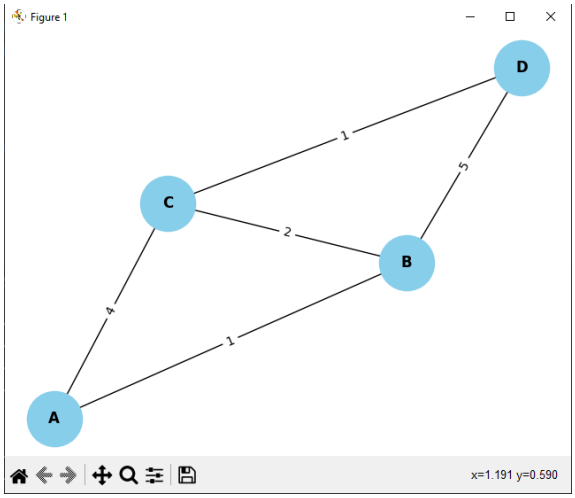
3 Classe Routing

La classe `Routing` si occupa di simulare il protocollo di routing e di rappresentare visivamente la rete.

- **Metodi**:
  - `routing_simulation(nodes, iterations=10)`: Simula il processo di aggiornamento delle tabelle di routing per un massimo di 10 iterazioni.
  - `draw_network(nodes)`: Disegna un grafo della rete, con nodi e connessioni (incluse le distanze) utilizzando `networkx` e `matplotlib`.

4 Funzionamento del sistema

Appena viene eseguito il programma, si apre una scheda con l'immagine del grafo della rete che contiene i nodi, rappresentati da dei cerchi azzurri, e gli archi che hanno il peso scritto come un numero in mezzo alla linea.



Appena viene chiusa la scheda dell'immagine, parte la simulazione che stampa le routing table di ogni nodo per ogni iterazione che esegue.

5 Requisiti per eseguire il codice

Per eseguire il codice sono necessari i moduli:

- **networkx**
- **matplotlib**

6 Link alla repository

github.com/NicolaGraziotin/DistanceVectorRouting