Report: Simulazione del Protocollo Distance Vector Routing

December 10, 2024

Introduzione

Questo report descrive l'implementazione e il funzionamento di un simulatore per il protocollo **Distance Vector Routing** in Python. Il programma simula una rete di nodi che calcolano dinamicamente le loro tabelle di routing utilizzando le informazioni condivise dai vicini.

Il protocollo **Distance Vector Routing** è un algoritmo distribuito utilizzato per determinare i percorsi più brevi tra nodi in una rete. Ogni nodo mantiene una tabella che associa ciascuna destinazione al costo e al prossimo hop del percorso più breve.

Funzionamento

Classe Node

Ogni nodo nella rete è rappresentato dalla classe `Node`. Essa include:

- `routing_table`: Una tabella che associa ciascuna destinazione al costo e al prossimo hop.
- `neighbors`: Un dizionario che memorizza i nodi vicini e i relativi costi di connessione.

Funzionalità principali

1. **Aggiunta di vicini**: Il metodo `add_neighbor` connette il nodo corrente a un vicino, aggiornando la tabella di routing iniziale.
2. **Aggiornamento della tabella di routing**: Il metodo `update_routing_table` implementa l'algoritmo Distance Vector,basato su Bellman Ford, aggiornando le rotte in base alle informazioni ricevute dai vicini.
3. **Simulazione**: La simulazione avviene iterativamente fino al raggiungimento della convergenza o al superamento di un numero massimo di iterazioni.

Inizializzazione da File

I nodi e le connessioni della rete sono definiti in un file di input. Il formato del file è il seguente:

```
NumeroNodi
Nodo1 Nodo2 Costo
Nodo1 Nodo3 Costo
...
```

Ad esempio, un file di input per una rete con 4 nodi potrebbe essere:

```
4
0 1 1
0 2 4
1 2 2
1 3 6
2 3 3
```

Output della Simulazione

Il programma stampa la tabella di routing di ciascun nodo a ogni iterazione. Di seguito un esempio di output:

```
=== Iterazione 1 ===
Routing Table for Node 0
Destination | Cost | Next Hop
     0      |  0   |    0
     1      |  1   |    1
     2      |  3   |    1
     3      |  6   |    1
------------------------------
...
Convergenza raggiunta!
```