

# Relazione Finale Progetto Python

Patrick Sbrighi 0001071460

Questo script implementa una simulazione di un protocollo di routing distribuito, basato sul concetto di Distance Vector Routing. Ogni nodo della rete mantiene una tabella di routing, ed essa viene aggiornata iterativamente comunicando con i nodi vicini. Lo script è strutturato in una classe Nodo ed un insieme di funzioni per la configurazione e simulazione della rete.

## 1. Struttura e funzionamento del codice

### 1.1 Classe Nodo

La classe Nodo rappresenta i nodi della rete. Questa è composta dai seguenti attributi e metodi:

- **Attributi:**
  - nome → il nome del nodo.
  - tabella → la tabella di routing, rappresentata come un dizionario che associa a ciascuna destinazione una tupla composta da distanza e next hop. Di default, la distanza verso sé stesso è 0 ed il next hop è sé stesso
  - vicini → un dizionario che elenca i nodi vicini e le rispettive distanze.
- **Metodi:**
  - aggiungiVicini(vicino, distanza) → aggiunge un nodo vicino specificandone la distanza e aggiorna la tabella per includerlo come destinazione direttamente raggiungibile.
  - aggiornaTabella() → aggiorna la tabella di routing confrontandola con le tabelle dei vicini. Se trova un percorso più breve o una nuova destinazione, aggiorna la tabella e ritorna True.
  - stampaTabella() → stampa la tabella di routing.

### 1.2 Funzione simulazione

Questa funzione rappresenta il ciclo principale della simulazione, infatti ad ogni iterazione viene stampato lo stato attuale delle tabelle di routing di tutti i nodi e ogni nodo aggiorna la propria tabella di routing basandosi sulle tabelle dei vicini.

La simulazione termina quando nessuna tabella viene più aggiornata perchè di fatto è stata raggiunta la convergenza del protocollo.

### 1.3 Configurazione della rete

La rete creata per la simulazione è composta da 7 nodi (NodoA a NodoG). I vicini di ciascun nodo e le rispettive distanze vengono configurati manualmente tramite il metodo per aggiungere i vicini.

(Per effettuare le varie prove sullo script vicini e distanze sono stati inseriti senza alcuna precisa logica)

## 2. Descrizione del protocollo Distance Vector Routing

Il DVR è un algoritmo distribuito in cui ogni nodo comunica periodicamente la propria tabella di routing ai vicini. Ogni nodo aggiorna la propria tabella in base a queste informazioni.

Il meccanismo si basa su:

- **Criterio di aggiornamento:** Se il nodo rileva un percorso più breve verso una destinazione aggiorna la propria tabella.
- **Convergenza:** Dopo un certo numero di iterazioni, tutte le tabelle dei nodi convergono e contengono i percorsi più brevi verso ogni altra destinazione.

## 3. Esecuzione dello script

Inizialmente ogni nodo conosce solo le distanze verso i propri vicini. Durante le varie iterazioni, ogni nodo scambia informazioni con i vicini e aggiorna la propria tabella nel caso in cui vengano trovati percorsi migliori. L'esecuzione termina quando nessuna tabella viene più aggiornata durante un iterazione, perchè questo significa che sono stati trovati tutti i percorsi migliori.

## 4. Output

Ogni iterazione stampa lo stato attuale delle tabelle di routing di ogni nodo. Alla fine, la simulazione stampa le tabelle finali, che per ogni riga indicano:

- La destinazione raggiungibile
- La distanza minima per raggiungere la destinazione
- Il next hop

## 5. Analisi del codice

La struttura del codice permette di mantenere separata la logica della rete dal comportamento di ciascun nodo e rende facile aggiungere nuovi nodi o modificare la topologia. Tuttavia sono attuabili alcuni miglioramenti, come per esempio:

- **Resilienza agli errori** → non è previsto il trattamento di nodi o collegamenti guasti.
- **Ottimizzazione** → la stampa delle tabelle potrebbe essere migliorata per visualizzare una topologia più complessa in modo compatto.