

Marco Buda  
Mat. 0001071464  
marco.buda3@studio.unibo.it

Traccia 2: Simulazione di Protocollo di Routing

# PastaDVR

Per la realizzazione del programma si è scelto di simulare in modo realistico seppur semplificato lo scambio delle informazioni descritto dal protocollo DVR. Con ciò si intende che le comunicazioni tra i vari nodi della rete sono state implementate per avvenire parallelamente (si è fatto uso del modulo `threading`), mentre i dettagli delle trasmissioni di livello IP sono stati omessi.

L'esecuzione del programma inizia con la creazione di una rete di router, come illustrato di seguito.

Figura 1: Schema della rete di default.

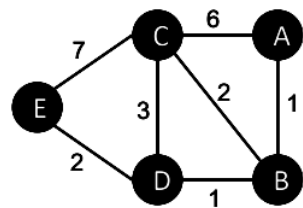


Figura 2: Raccolta delle informazioni in una tabella.

	B	C	D	E
A	1	6	/	/
B		2	1	/
C			3	7
D				2

Snippet 1: Traduzione dei dati in codice.

```
155 Network(  
156     ['A', 'B', 'C', 'D', 'E'],  
157     [1, 6, None, None, 2, 1, None, 3, 7, 2]
```

Il passo successivo è invocare il metodo `dvr_start` della rete appena creata. Questo segnalerà a tutti i router  $s$  di avviare la sequenza di start, che consiste nelle seguenti fasi:

- Inizializzazione del proprio DV con valore iniziale  $\{s \rightarrow (\text{cost} = 0)\}$ :

```
66 self._dv[self._name] = (0, None)
```

- Scoperta dei nodi vicini  $j$  e memorizzazione del costo dei collegamenti diretti nel proprio DV:  $j \rightarrow (\text{cost} = d_{sj}, \text{next hop} = j)$ :

```
77 for link in self._links:
78     cost, neighbor_name = link.get_info()
79     self._dv[neighbor_name] = (cost, neighbor_name)
```

- Invio del proprio DV attuale a tutti i nodi vicini:

```
82 for link in self._links:
83     link.send((self._name, self._dv))
```

Successivamente i router si mettono in ascolto. Quando ricevono il DV di un nodo vicino, ne esaminano le voci e, qualora trovino dei percorsi verso nodi sconosciuti, oppure percorsi di costo inferiore alla stima attuale, li memorizzano nel proprio DV, il tutto tenendo conto del costo del collegamento al nodo vicino.

```
100 if node_name not in self._dv or link_cost + cost < self._dv[node_name][0]:
101     self._dv[node_name] = (link_cost + cost, sender)
```

Se il DV è stato modificato, il router lo inoltra a tutti i nodi vicini.

Il programma termina automaticamente quando i DV di tutti i router hanno raggiunto l'ottimalità, ossia i nodi hanno terminato di scambiarsi messaggi.

### Esecuzione del programma:

Per avviare lo script è necessario avere installato Python 3.9+ ed eseguire il comando:  
`python dvr.py`

Di default viene utilizzata la rete riportata alla slide 21 del file `2_Routing_new.pdf` condiviso su Virtuale, ma è possibile modificarla direttamente dal codice.

Viene fornita anche un'altra rete che richiede solo di essere decommentata.

Durante l'esecuzione, vengono stampati a video i DV dei vari router ogni volta che vengono modificati.

Si allega uno screenshot che mostra i DV ottimali ottenuti al termine del processo sulla rete di default.

Per simulare il ritardo introdotto dai collegamenti si è scelto di interrompere i thread associati per un numero di secondi equivalente al costo, utilizzando il metodo `time.sleep`. Per reti personalizzate potrebbe essere preferibile rimuovere questa funzionalità.