



Relazione elaborato Fondamenti di Reti di Telecomunicazioni

Niccolò Piazzesi
Anno Accademico 2019/20

Contents

1	Introduzione	3
1.1	Prerequisiti	3
1.2	Struttura del codice	3
2	Implementazione	4
2.1	Soluzione Proposta	4
2.2	Costruzione della rete e implementazione dei passaggi dell'algoritmo	4
2.3	Scelte implementative	5

1 Introduzione

1.1 Prerequisiti

Il progetto è stato sviluppato in Python, utilizzando la versione 3.8.1 del linguaggio. Il codice è retrocompatibile con le vecchie versioni di Python 3, mentre presenta dei problemi quando viene eseguito con Python 2. Si consiglia quindi di utilizzare esclusivamente Python 3.

1.2 Struttura del codice

Il codice è stato suddiviso in 5 moduli. Nei moduli **simple_test** e **25_test** è stata verificata la soluzione proposta.

In particolare, in **simple_test** è stata utilizzata una rete composta da 5 nodi con topologia a maglia e 5 dispositivi.

In **25_test** invece è stata utilizzata la rete con 25 nodi e 25 dispositivi richiesta. La topologia scelta è stata quella ad anello e ogni dispositivo è stato posizionato tra due nodi successivi. Per semplicità, in 25_test la priorità dei dispositivi viene scelta in modo casuale ad ogni esecuzione.

Per eseguire i due moduli, si deve aprire un terminale nella cartella del progetto ed eseguire i seguenti comandi:

```
python3 simple_test.py
```

```
python3 25_test.py
```

2 Implementazione

2.1 Soluzione Proposta

La soluzione è definita nella funzione **stable_marriage** del modulo **marriages** ed è basata sull'algoritmo di Gale-Shapley. Questa funzione prende in input una lista di nodi e una lista di dispositivi e li associa attraverso i seguenti passaggi, ripetuti fino a quando tutti i dispositivi sono associati a un nodo:

- Ciascun dispositivo non ancora assegnato si propone al nodo più vicino tra quelli a cui non si è ancora proposto.
- Ciascun nodo che ha ricevuto almeno una proposta:
 - si associa al dispositivo con priorità maggiore tra quelli che si sono proposti se è libero, oppure se esso ha priorità maggiore rispetto al dispositivo con il quale è già associato (che verrà liberato). In caso contrario, rifiuta la proposta.
 - rifiuta, se presenti, le proposte rimanenti

2.2 Costruzione della rete e implementazione dei passaggi dell'algoritmo

Nel modulo **Network** sono definite le classi **Device**, **Node**, **Network**.

Una rete è costruita come un'istanza della classe **Network** a partire da una lista di nodi. In **Network** vengono inoltre definiti i metodi per aggiungere e rimuovere archi.

Ogni nodo mantiene i dispositivi che si sono proposti in un insieme detto **proposals** e effettua la sua scelta attraverso **choose_device**.

Un dispositivo viene istanziato con una determinata priorità. Mantiene la lista dei nodi ordinata per vicinanza in **ranked_nodes**. Questa lista viene costruita nel metodo **position_device**. Partendo dai vicini, ogni dispositivo effettua una visita in ampiezza della rete e inserisce i nodi nell'ordine in cui li incontra. Ogni dispositivo si propone al nodo preferito attraverso **propose**.

2.3 Scelte implementative

Poiché la lista di preferenza dei dispositivi è unica per tutti i nodi è stato scelto di non memorizzarla. Al posto di controllare una lista ordinata, ad ogni iterazione ogni nodo si associa al dispositivo con priorità massima tra quelli che può scegliere.