

Michele Sprocatti 2003803

Report Homework 2 Reti di Calcolatori parte Multimedia

1. Descrizione su come lanciare il codice (sistema operativo, linguaggio di scripting etc.)

Il codice è riassunto tutto in un unico script scritto in un notebook python, la scelta è stata fatta per avere gli output vicini al codice e per la possibilità di separare le varie sezioni dello script con commenti e titoli.

Lo script è stato eseguito su visual studio code installato su un computer con sistema operativo MacOS Monterey in cui è installato python 3.9.6. Per l'esecuzione, quindi, serve aver installato python nel proprio dispositivo.

Inoltre, è necessario avere installato le librerie python necessarie che sono numpy, matplotlib e seaborn:

- *numpy* è stato usato per effettuare le operazioni matematiche
- *matplotlib* è stato usato per plottare i grafici
- *seaborn* è stato utilizzato per disegnare il grafico relativo a come varia l'RTT minimo in funzione di L, questo per avere il risultato di una regressione lineare

L'utilizzo di queste tre librerie prevede che siano eseguite sempre le prime 4 celle del notebook:

- La prima cella del notebook fa tutti gli import che serviranno al programma
- La seconda dove si definisce la funzione per trovare il minimo TTL necessario per arrivare ad una destinazione tramite una serie di comandi ping
- La terza che definisce la funzione per trovare il minimo TTL necessario per arrivare ad una destinazione tramite il comando traceroute
- La quarta dove ci sono tutte le variabili necessarie per effettuare i conti successivi e si calcola il TTL minimo nei due modi tramite l'utilizzo delle due funzioni

Questo per evitare eventuali errori nel caso che la cella che si voglia eseguire sfrutti una funzione di queste librerie o usi le variabili definite nella quarta.

Il codice genera due file di output se non sono già presenti altrimenti sovrascrive il loro contenuto:

- *resTab.txt*: contiene le varie informazioni riguardo ai tempi delle varie sessioni di ping. In particolare, è stato organizzato come un file csv in cui ogni una riga corrisponde ad una sessione di ping e contiene i valori riguardanti RTTmin, RTTavg, RTTmax, RTTstd della sessione stessa.
- *res.txt*: contiene l'output di tutti i comandi ping che sono stati eseguiti durante lo script.

2. Descrizione dei parametri dell'esperimento: server, numero di istanze K, dimensioni dei pacchetti, altre (eventuali opzioni del ping, etc.)

È stato scelto il server *atl.speedtest.clouvider.net* corrispondente alla città di Atlanta, è comunque possibile modificare la scelta cambiando solo il valore assegnato alla variabile server all'interno del codice. Per quanto riguarda la scelta della dimensione dei pacchetti si usa la funzione range di python con i seguenti valori:

Lb=range (29,1450,29)

La scelta del range in cui variare la lunghezza in byte porta ad avere 50 sessioni di ping diverse. Per ogni sessione il programma invia 8 pacchetti diversi(K).

Le opzioni del comando ping che sono state usate sono:

- -c per specificare il numero di pacchetti da inviare.
- -s per specificare la dimensione in byte del pacchetto, a questa dimensione poi bisognerà aggiungere 28 byte per avere la dimensione reale del pacchetto inviato (8 byte di ICMP e 20 byte di IP).
- -m per specificare il TTL (time to live) dei pacchetti.

Da notare che i flag del comando ping sono diversi a seconda del sistema operativo quindi se si esegue lo script su computer dove c'è installato Windows o Linux bisogna prestare attenzione ai flag ed eventualmente correggerli all'interno dello script per una corretta esecuzione dello stesso.

3. Stima del numero di link attraversati

La stima del numero di link attraversati viene fatta attraverso ping successivi con un TTL incrementale fino a quando non c'è più un 100.0% di packet loss rate. Il risultato della seguente stima viene poi verificato eseguendo il comando traceroute e solo se i due risultati sono uguali il programma effettua le varie sessioni di ping, questa scelta deriva dal fatto che i due risultati non possono essere diversi se si eseguono entrambe le stime nella stessa rete.

4. Andamento dell'RTT minimo, medio, massimo e della deviazione standard in funzione della dimensione del pacchetto

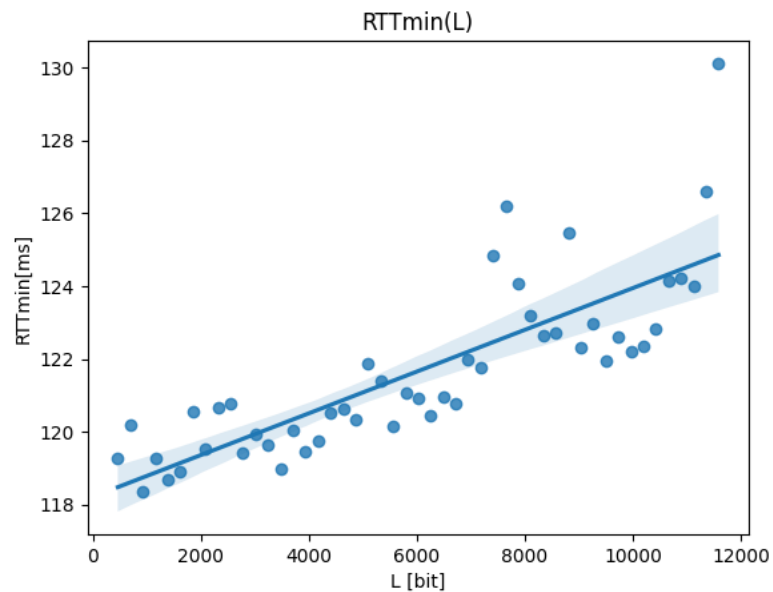


Figura 1: RTT minimo in funzione della lunghezza espressa in bit

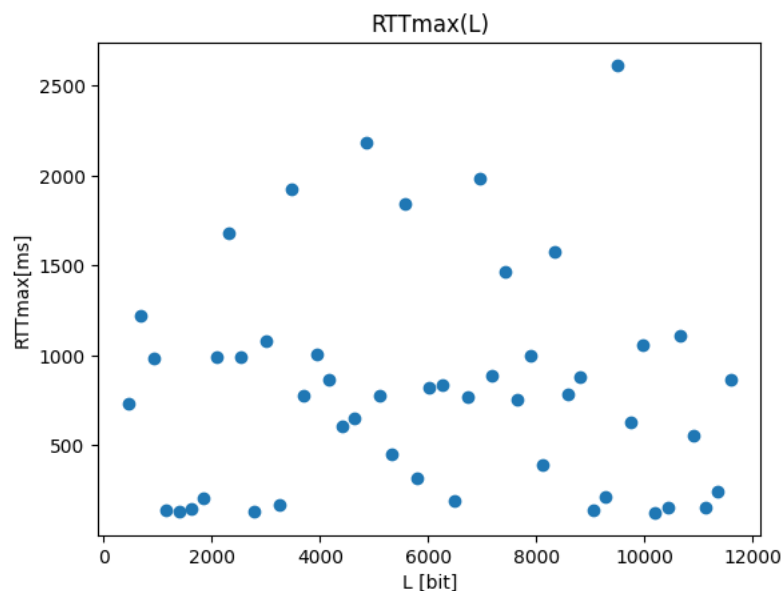


Figura 2: RTT massimo in funzione della lunghezza espressa in bit

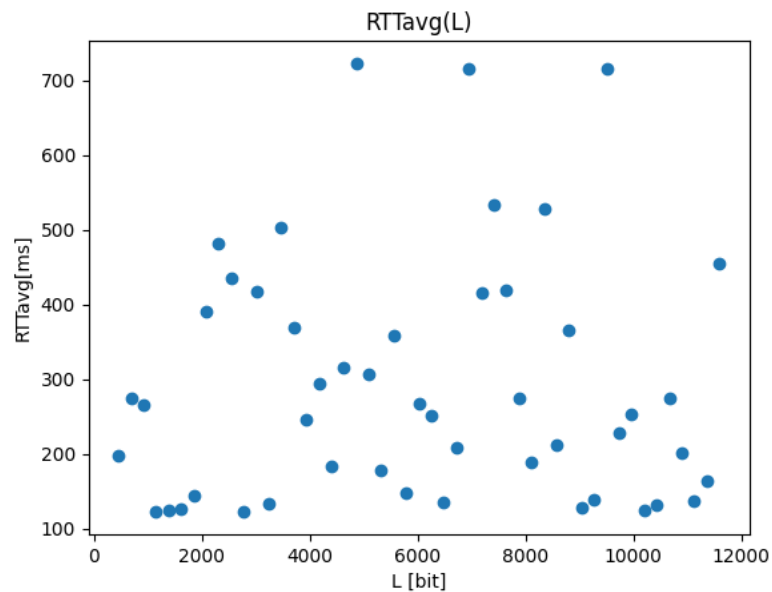


Figura 3: RTT medio in funzione della lunghezza espressa in bit

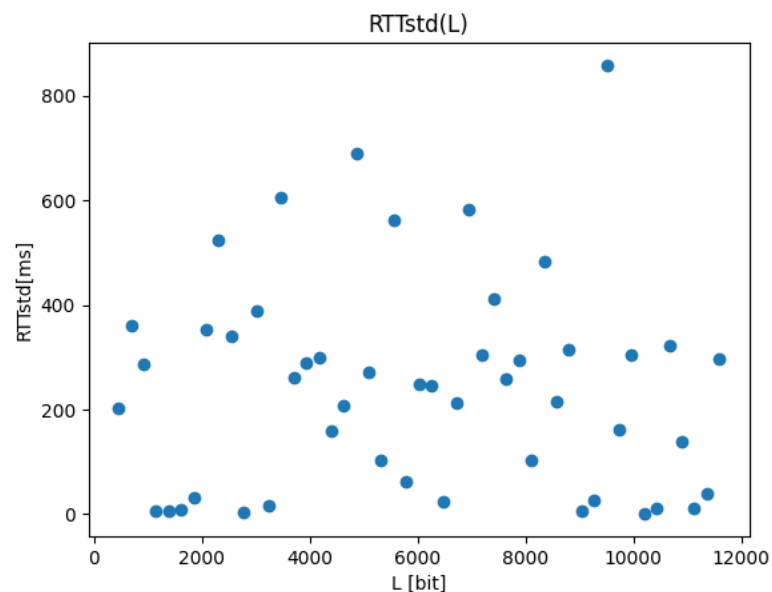


Figura 4: Deviazione standard dell'RTT in funzione della lunghezza espressa in bit

Dato il grafico si può confermare che l'RTTmin cresce al crescere della lunghezza dei pacchetti, questo deriva dal fatto che i ritardi dei vari passi della trasmissione sono più lunghi vista la dimensione maggiore del pacchetto. Infatti, il risultato dell'interpolazione è una retta con coefficiente angolare positivo.

Per quanto riguarda RTTmax si può vedere che è molto variabile e questo può dipendere da eventuali ritardi nei nodi attraversati dai pacchetti.

Il grafico di RTTavg fa notare che la media della singola sessione varia molto se si considera l'insieme delle varie sessioni.

Infine, il grafico di RTTstd fa notare come l'RTT sia molto variabile all'interno della singola sessione di ping.

5. Stima di R e di $R_{bottleneck}$

Per iniziare è stato stimato il valore di \underline{g} data la formula:

$$RTT_{min} = aL + T$$

Per eseguire la stima è stata usata la funzione di `numpy.polynomial.polynomial.Polynomial.fit`. Questa funzione è stata usata seguendo la documentazione di numpy con la seguente istruzione:

`T, a=np.polynomial.polynomial.Polynomial.fit((Lb+28) * 8, min_time, 1).convert().coef`

Per stimare R e Rbottleneck sono state usate le formule date nel testo dell'homework e i risultati sono i seguenti:

- Se supponiamo che tutti i link abbiano lo stesso throughput si ha:

$$R = \frac{n}{a} = 45423.43186900169 \text{ b/ms} = 45423431.86900169 \text{ b/s}$$

- Se invece supponiamo che ci sia stato un link che abbia un throughput molto minore e quindi si sia comportato come bottleneck si ha:

$$R_b = \frac{2}{a} = 3494.1101437693606 \text{ b/ms} = 3494110.1437693606 \text{ b/s}$$

6. Discussione dei risultati ottenuti

Dati questi risultati si ipotizza che la stima migliore del throughput sia quella che risulta dall'ipotesi che ci sia stato un nodo più lento degli altri, questo perché i vari server attraverso cui si passa sono molto più performanti, effettuando ricerche si può vedere come abbiano velocità nell'ordine di Gbps, rispetto alla rete locale a cui è stato collegato il computer durante i test, che da contratto con l'ISP prevede un massimo di 100 Mbps, ma che nella realtà è molto meno performante; quindi in conclusione la rete locale con cui si sono fatti i test è stata la bottleneck rispetto al percorso completo che fanno i vari pacchetti. A supporto di questo ragionamento è stato eseguito il programma di iPerf3 che permette di misurare la massima banda di una rete IP, il risultato di questo strumento è molto simile al risultato ottenuto, sotto l'ipotesi della bottleneck, tramite lo script python.