COME LANCIARE IL CODICE

Questo homework è stato realizzato su Windows 11 e scritto su Python, linguaggio scelto per la facilità con cui la sua libreria *os* permette di eseguire i comandi di ping. Inoltre, come comando per eseguire i test è stato usato sia il comando preinstallato *ping*, per la stima dei link attraversati, che il comando installabile *psping*, per il calcolo degli RTT.

Pertanto, il primo passaggio per eseguire il codice è assicurarsi di usare Windows 11 e installare *psping*. Per farlo:

- andare su questo link
- cliccare su *Download PsTools*
- nel file zip scaricato, individuare i file *psping.exe* e *psping64.exe*
- individuare la cartella contenente l'eseguibile del prompt dei comandi, *cmd.exe* (di norma la cartella è *C:\Windows\System32*)
- estrarre i due file citati sopra in questa cartella

Successivamente, bisogna avere Python installato sulla propria macchina e installare due librerie esterne utilizzate nel codice con il comando da riga di comando *pip install <nome-libreria>*, se non già installate tramite software come Anaconda:

- *numpy*, usata per ricavare i valori particolari degli RTT (minimo, massimo, media, deviazione standard)
- *matplotlib*, usata per rappresentare graficamente gli andamenti dei valori

Una volta svolti questi preparativi, è possibile eseguire il codice sia dal prompt dei comandi di Windows che dall'IDLE di Python. Se si sceglie il prompt dei comandi, è necessario navigare fino alla cartella in cui è stato estratto il contenuto del file zip ed eseguire il comando *python homework2.py*. Su Windows 11, è possibile anche aprire la cartella con Esplora Risorse, fare clic destro sullo spazio vuoto (non su un file) e scegliere "Apri nel terminale".

Una volta lanciato il codice, si ha la possibilità di salvare tutto l'output prodotto dal file in un file di testo; se si sceglie altrimenti, l'output verrà stampato a schermo. Indipendentemente dall'opzione scelta, *psping* stamperà a schermo dei warning ogni volta che una richiesta di ping finisce con un timeout. Se succede non serve chiudere o riavviare il programma, visto che gestisce questi casi e fa continuare il programma ripetendo la richiesta fallita. Verso la fine del programma, verranno aperti i grafici prodotti. Il programma entra in standby dopo l'apertura di ognuno di essi, aspettano che vengano chiusi. Una volta chiusi tutti e quattro i grafici, il programma termina.

DESCRIZIONE DEI PARAMETRI

- Comandi usati per ping: *ping, psping*
- Server usato: *paris.testdebit.info*
- Numero di istanze *K*: 20
- Dimensioni dei pacchetti: intervallo [50,1420], valori presi ad intervalli di 50

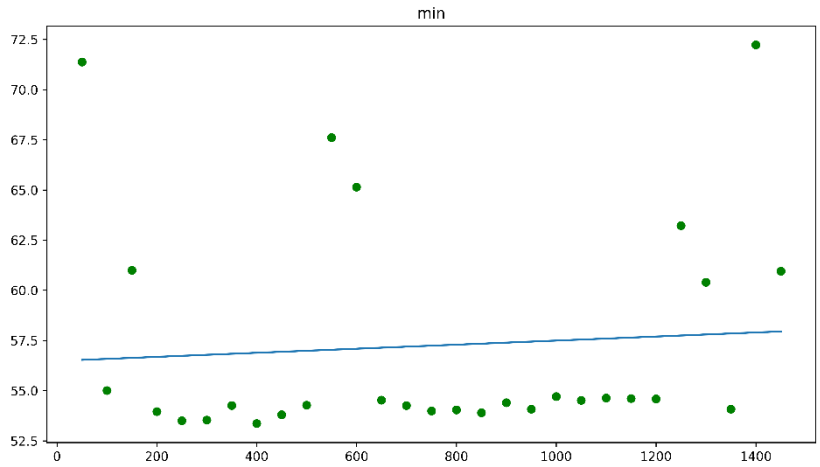
STIMA DEL NUMERO DI LINK ATTRAVERSATI

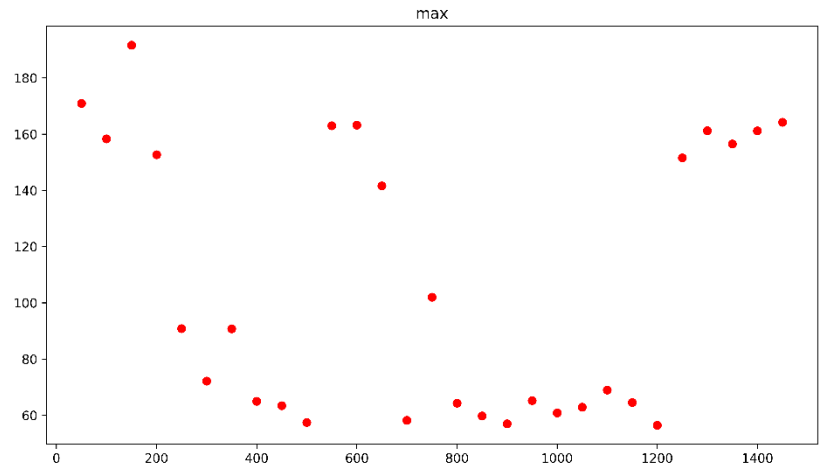
- Usando *psping*: 11
- Usando *tracert*: 11

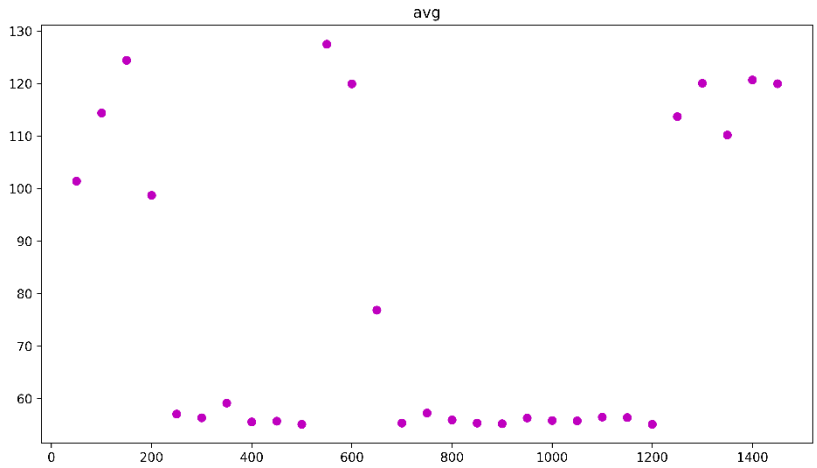
```
TTL in uso: 11
Con ping: 11
tracert in uso
Con tracert: 11
```

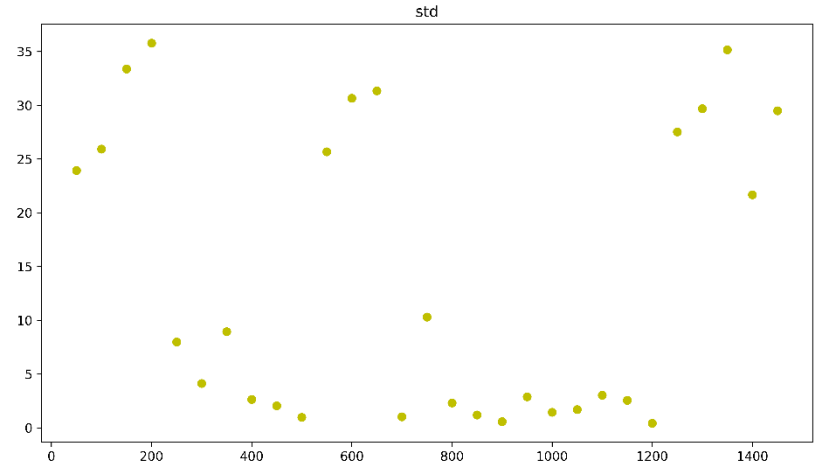
ANDAMENTO DELL'RTT MINIMO, MEDIO, MASSIMO E DELLA DEVIAZIONE STANDARD IN FUNZIONE DELLA DIMENSIONE DEL PACCHETTO

In tutti i grafici, le ascisse rappresentano la dimensione del pacchetto in byte e le ordinate rappresentano l'RTT in millisecondi.









STIMA DI S E $S_{BOTTLENECK}$

- S = 21734 bit/s
- $S_{bottleneck}$ = 1975 bit/s

```
EQUAZIONE DELLA FUNZIONE L-RTT = 57.2403448275862 + 0.7085517241379379 x**1
A = 0.0010122167487684826
S = 21734.475374732127 bit/s
S_BOTTLENECK = 1975.8613977029206 bit/s
```

DISCUSSIONE DEI RISULTATI OTTENUTI

Un primo fenomeno che si è verificato nell'insieme dei test effettuati è la grande varianza dei throughput trovati da test a test, che spaziavano da valori più grandi di quello riportato di un ordine di grandezza circa a valori negativi. Questa variabilità delle misure potrebbe derivare dalla serie di semplificazioni effettuate per ottenere le formule usate per calcolare i throughput.

Una costante di tutti i test, tuttavia, è quanto sia ridotta la pendenza dell'interpolazione lineare tra L e RTT minimi, ovvero quanto crescano lentamente gli RTT all'aumentare di L. Ciò significa che il contributo agli RTT

del coefficiente *a*, ovvero delle durate dei vari segnali, è molto minore rispetto all'intercetta *T*, ovvero dei tempi di propagazione dei segnali nei vari mezzi che attraversano.

È anche interessante notare come effettuando test su server che si trovano oltreoceano, oltre ad avere RTT maggiori, i throughput in media aumentino sensibilmente, anche di 10 volte. Usando *nyc.speedtest.clouvider.net* e lasciando invariati gli altri parametri si possono infatti ottenere valori intorno ai 150.000 bit/s. Questo fenomeno è probabilmente dovuto all'utilizzo dei cavi sottomarini localizzati nell'Oceano Atlantico, che presentano capacità nominali nell'ordine dei giga/terabit al secondo (dati presi da https://it.wikipedia.org/wiki/Transatlantic_Communications_Cable).