

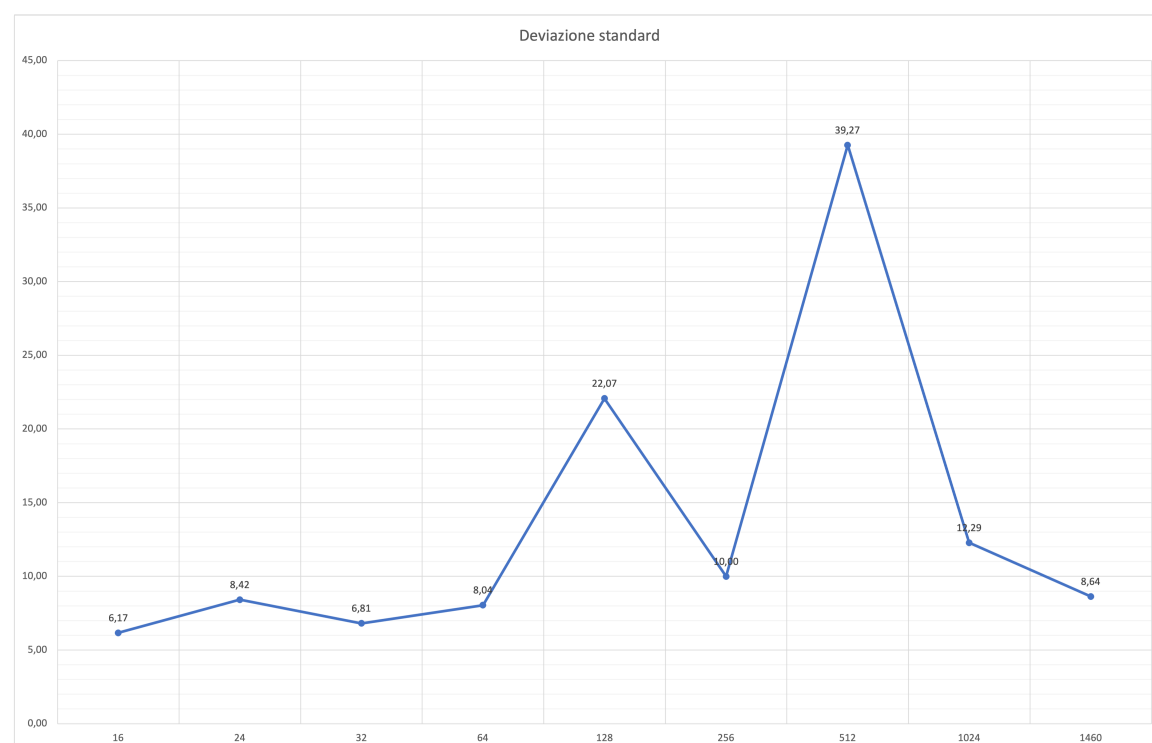
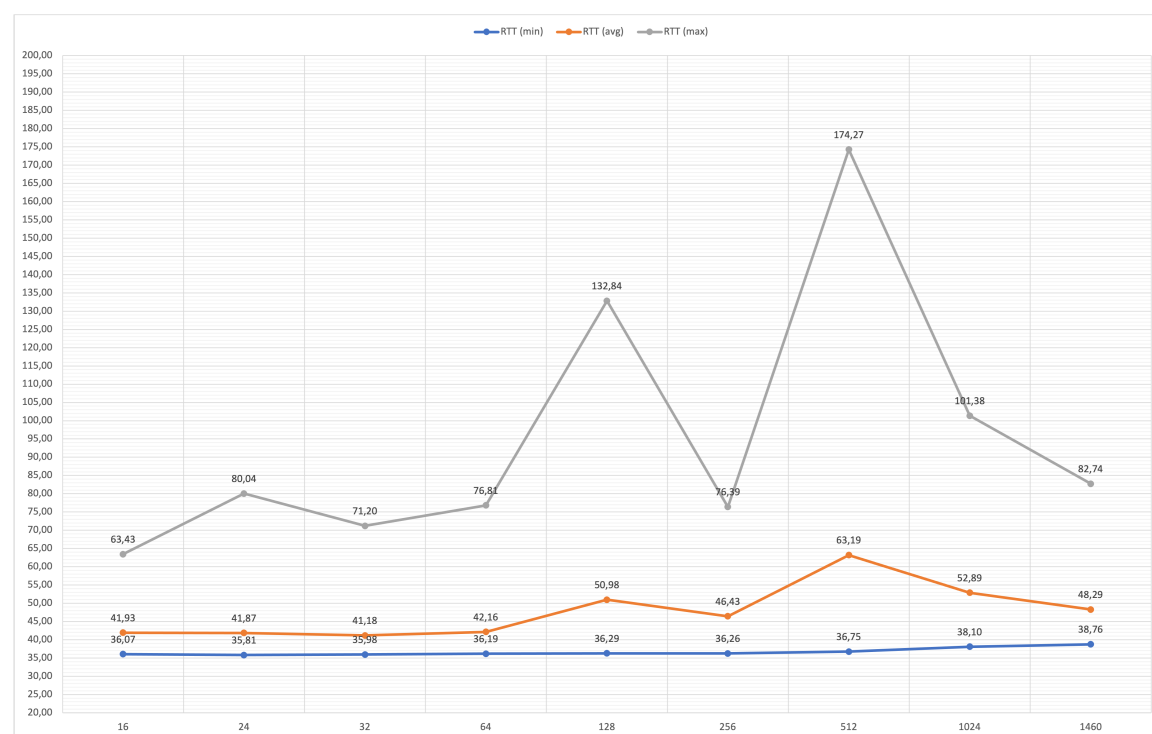
Network Performance

Simone Peraro 1216334

Il nostro scopo è valutare le performance di rete utilizzando i comandi *ping* e *iperf*.

Innanzitutto verifichiamo il numero di hop presenti tra la sorgente e la destinazione: utilizziamo il comando *ping* con il flag *-m* per determinare il numero di hop necessari a raggiungere il server di destinazione, che in questo caso si tratta dell'host *speedtest.serverius.net*. Dopo alcuni tentativi, è facile capire che il numero di hop tra la sorgente e la destinazione è pari a 13.

A questo punto valutiamo il *Round Trip Time* dei pacchetti, utilizzando il flag *-s* per specificare il payload del pacchetto *ICMP ECHO REQUEST*. Utilizziamo le seguenti quantità di byte: 16, 24, 32, 64, 128, 256, 512, 1024, 1460 in combinazione con il flag *-D* per evitare che i pacchetti troppo grandi vengano frammentati. Ricordiamo che ad ognuno di questi payload vengono aggiunti 8 byte di header del protocollo *ICMP* e altri 20 byte di header del protocollo *IP*. A questo punto possiamo graficare i valori minimo, medio e massimo dell'*RTT*.



Al variare della dimensione del payload si vede un leggero aumento del RTT minimo: questo è dovuto soprattutto al ritardo di accodamento dei pacchetti che diventano sempre più grandi e quindi tendono a saturare i buffer dei vari nodi lungo il percorso.

Proviamo ora a calcolare il bitrate del nodo più lento. Sappiamo che

$$RTT = \sum \frac{L}{R_i} + q_i$$

Dobbiamo trovare il modo di eliminare il tempo speso dal pacchetto nel buffer di accodamento. Intercettando i valori di *RTT* minimo con l'asse delle ordinate (cioè simulando un pacchetto con payload nullo), possiamo calcolare in modo approssimativo il valore medio di q_i , che in questo caso risulta essere circa di $35,86ms$. Calcolato questo valore, possiamo effettuare una serie di

richieste *ICMP* verso la sorgente con un payload contenuto, in modo da non rendere troppo evidenti gli effetti dell'accodamento. Inviando una serie di richieste con payload pari a 16 byte, otteniamo un *RTT* minimo pari a $38,44ms$. Quindi eliminando il ritardo totale di accodamento, otteniamo un tempo di latenza dovuto allo scambio di bit tra i nodi pari a $2,58ms$. Ora possiamo provare a calcolare R^* : sappiamo infatti che

$$Delay = \sum \frac{L}{R_i}$$

tuttavia, se consideriamo che un nodo k ha il bitrate minimo $R_{i_{min}}$, allora la somma dei fattori $\frac{L}{R_i}$ può essere approssimativamente maggiorata dal singolo fattore $\frac{L}{R_{i_{min}}}$. Pertanto possiamo ricavare direttamente il bitrate minimo R^* invertendo la formula.

$$R^* = \frac{L_{IP} * 2 * n_{hop}}{Delay}$$

dove L_{IP} è la lunghezza del pacchetto contando i bit fino al livello IP mentre n_{hop} è il numero di nodi tra la sorgente e la destinazione (ma bisogna considerare che l'*RTT* conta il tempo totale di andata e ritorno). Inserendo i valori e convertendo il risultato in Mb/s otteniamo un bitrate di circa $7,42 Mb/s$.

Ora possiamo provare a effettuare un test tramite il comando *iperf* per verificare l'effettiva velocità di connessione

```
Connecting to host speedtest.serverius.net, port 5002
[ 6] local 192.168.1.208 port 52037 connected to 178.21.16.76 port 5002
[ ID] Interval      Transfer    Bandwidth
[ 6]  0.00-1.00  sec  3.94 MBytes  32.9 Mbits/sec
[ 6]  1.00-2.00  sec  1.25 MBytes  10.5 Mbits/sec
[ 6]  2.00-3.01  sec  0.00 Bytes   0.00 bits/sec
[ 6]  3.01-4.01  sec  0.00 Bytes   0.00 bits/sec
[ 6]  4.01-5.00  sec  32.6 KBytes  268 Kbits/sec
[ 6]  5.00-6.00  sec   866 KBytes  7.07 Mbits/sec
[ 6]  6.00-7.00  sec  1.32 MBytes  11.1 Mbits/sec
[ 6]  7.00-8.00  sec  1.50 MBytes  12.6 Mbits/sec
[ 6]  8.00-9.00  sec  1.84 MBytes  15.4 Mbits/sec
[ 6]  9.00-10.00 sec  2.11 MBytes  17.7 Mbits/sec
- - - - -
[ ID] Interval      Transfer    Bandwidth
[ 6]  0.00-10.00 sec  12.8 MBytes  10.8 Mbits/sec  sender
[ 6]  0.00-10.00 sec  12.6 MBytes  10.6 Mbits/sec  receiver
```

Si nota come il valore ottenuto dal calcolo tramite *RTT* non è così distante dal risultato ottenuto su *iperf* considerando anche il fatto che il protocollo *ICMP* è un protocollo solitamente a bassa priorità rispetto ai normali pacchetti *TCP/UDP* presenti nella rete. Bisogna considerare inoltre che il test è stato condotto in condizioni poco controllate ed è molto sensibile alle variazioni nella rete (dispositivo connesso tramite bridge wifi, presenza di dispositivi smart che disturbano le normali condizioni della rete ecc...)