

LABORATORIO DI INTERNET



**Politecnico
di Torino**

Report II

Stima di velocità tramite ping

Gruppo **14**

Andreas Brummer (s270332)

Alessandro Ciullo (s269589)

Andrea Scamporrino (s270971)

1 Configurazione di rete

Nell'effettuare i test é stata usata la seguente configurazione di rete:

Indirizzo di rete	172.16.14.0
Indirizzo di broadcast	172.16.14.63
NetMask	255.255.255.192



Nome host	Indirizzo IP
H1	172.16.14.1/26
H2	172.16.14.2/26

Tabella 1.1: Configurazione di rete

2 Considerazioni sui parametri e le configurazioni utilizzate

Nei vari scenari analizzati abbiamo utilizzato i 2 computer fissi del laboratorio collegati alla rete tramite cavo ethernet. Inoltre, abbiamo utilizzato degli adattatori usb-ethernet per la connessione perchè senza di essi la scheda di rete del computer svolge delle operazioni aggiuntive che interferiscono con i test svolti.

2.1 RTT

Nei vari calcoli del RTT si ipotizzerà che sia i tempi di elaborazione che i tempi di propagazione siano trascurabili. Le formule reali dovrebbero essere del tipo

$$RTT_{reale} = RTT_{TX} + \eta \quad (1)$$

Il primo termine rappresenta la somma di tutti i tempi di trasmissione, mentre il secondo indica un effetto di rumore dato dai tempi di elaborazione, propagazione e anche da altri fenomeni non deterministici. Per minimizzare il contributo di η teniamo conto, in ciascuno dei test, del RTT minimo, misurato al variare della dimensione dei pacchetti inviati..

2.2 Frammentazione

Quando comunichiamo attraverso un canale ethernet la Maximum Transmission Unit (MTU) è di 1500 byte. Per ovviare a questo limite il protocollo IP, consapevole della MTU del livello di collegamenti dati su cui si appoggia, svolge un'operazione chiamata "frammentazione". L'IP fragmentation scompone le SDU in unità più piccole adatte alla trasmissione sul canale, e utilizza i campi "Fragment offset", "Don't Fragment" e "More Fragments" per permettere ai dispositivi di livello di rete di riassemblare il pacchetto alla loro ricezione. Il fenomeno della frammentazione ha alcune importanti ripercussioni sul comportamento del RTT e della capacità al variare di S durante l'inoltro di pacchetti sulla rete internet. Nel momento in cui un nuovo frammento viene generato il numero di dati da trasmettere dall'host incrementa in maniera non lineare: come mostrato in figura 2.1 prima si verifica un salto e in seguito un plateau che perdura finchè viene aggiunto del padding (necessario al raggiungimento del minimum frame size) al nuovo pacchetto.

2.3 Calcolo di D(S)

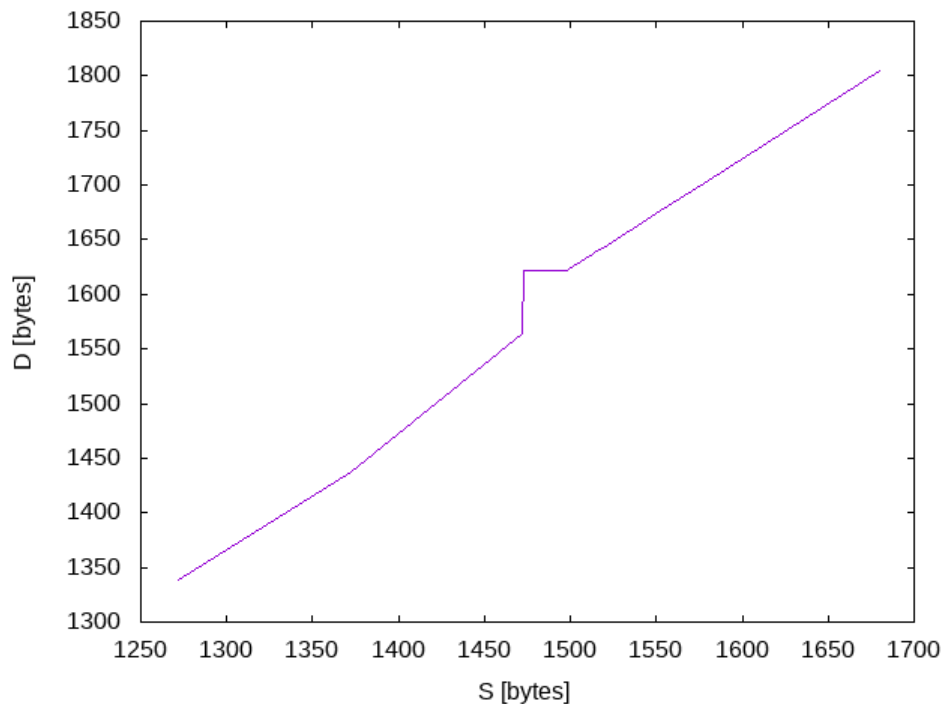


Figura 2.1: Grafico S-D

I dati trasmessi a livello fisico (D) sono ottenuti come la somma tra i dati di livello applicazione (S) e gli header ICMP (8 Byte), IP (20 Byte), Ethernet (26 Byte), un eventuale padding, ed un interpacket gap (12 Byte). Possiamo esprimere D(s) in maniera rigorosa con il seguente sistema:

$$\begin{cases} D(S) = S + H_{ICMP} + H_{IP} + H_{ETH}, S \notin [1473 + 1472n, 1497 + 1472n], n \in \mathbf{N} \\ D(S) = S + H_{ICMP} + H_{IP} + H_{ETH} + 46 - (((S + 8) \% 1480) + H_{IP}), S \in [1473 + 1472n, 1497 + 1472n], n \in \mathbf{N} \end{cases}$$

3 Host collegati direttamente tramite cavo ethernet

In questa sezione analizziamo lo scenario in cui i due host H1 e H2 sono collegati in maniera diretta mediante un cavo Ethernet. Il modello utilizzato (fig. 4.2) è semplice poichè $RTT = 2 \cdot T_{TX} = \frac{2D}{C}$. Ricaviamo quindi che

$$C = \frac{2D}{RTT} \quad (2)$$

3.1 Velocità: 10Mb/s

Abbiamo imposto la velocità di trasmissione sul canale a 10Mb/s Full Duplex eseguendo il seguente comando su uno dei due host.

```
1 sudo ethtool -s eth0 speed 10 duplex full autoneg on
```

Attraverso il processo di auto-negoziazione viene impostata la velocità su entrambi gli host.

In figura 4.3 il RTT teorico è messo a confronto con il RTT misurato con l'ausilio dello script 4.11. Essendo la velocità di trasferimento relativamente bassa, i tempi di elaborazione e propagazione diventano trascurabili rispetto al tempo di trasmissione e perciò ne consegue un comportamento reale estremamente simile a quello teorico. Seppur difficile da notare, per via della scala sulle ascisse molto ampia, è importante prestare attenzione al comportamento nell'intorno dei punti di frammentazione: il RTT compie un salto verso l'alto ad ogni frammentazione causato dal repentino aumento di D, e resta costante per il "periodo di padding" che consegue, continua poi il suo andamento lineare fino al prossimo punto di frammentazione.

In figura 4.4 è invece riportato il grafico della capacità sul canale calcolata mediante l'equazione 2. La funzione ha un andamento crescente e come potevamo immaginare per $S \rightarrow \infty$ la capacità tende al valore negoziato tra i due host in maniera asintotica. In corrispondenza dei punti di frammentazione abbiamo un comportamento difficile da analizzare a causa di fattori fuori dalla nostra comprensione (tempi di elaborazione, operazioni svolte dalla scheda di rete ecc.) e abbiamo perciò potuto solo avanzare delle ipotesi: Come mostrato in appendice la capacità aumenta repentinamente nel primo Byte successivo alla frammentazione e poi decresce per un breve intervallo; al termine del fenomeno la funzione riassume il suo andamento naturale fino al successivo punto di frammentazione. ↖ NO

Tenendo conto della formula $C = D/(2T_{TX}(D) + \eta)$ e del fatto che il tempo di trasmissione vari in maniera lineare in funzione di D, possiamo ipotizzare che ad un aumento improvviso di D il contributo del termine η si riduca e ne consegua un brusco aumento di C. Abbiamo, inoltre, immaginato che il successivo andamento decrescente avvenga nell'intervallo di padding e che sia causato dall'aumento dei tempi di elaborazione: seppur il valore D non cambi i dati elaborati a livello applicazione aumentano generando dei costi computazionali; l'inserimento del padding, invece, è fatto in maniera efficiente dalla scheda di rete e non comporta costi addizionali. Ne consegue perciò un andamento locale decrescente.

3.2 Velocità: 100Mb/s

In questa sezione abbiamo configurato il bit-rate del canale a 100Mbps. Come osservabile nell'immagine 4.5, l'andamento della funzione è irregolare. L'impatto di η , e quindi dei fenomeni non deterministici, è molto maggior poiché la velocità di trasmissione è estremamente alta, ed inoltre l'imprecisione delle misure temporali diventa molto più evidente su una scala dei tempi così ridotta. L'offset ($\simeq 0,5ms$) tra il RTT misurato e il RTT teorico è causato dai tempi di elaborazione. Questa divergenza, seppur all'apparenza costante, aumenta al crescere dei dati trasmessi.

La capacità, mostrata in figura 4.6, ha anche questa volta un andamento crescente con asintoto orizzontale a 100Mbps, ed in presenza di frammentazione si comporta in maniera simile al caso mostrato al punto 3.1, a meno di una piccola differenza: questa volta l'incremento repentino di C è meno evidente e prevale l'andamento decrescente nell'intorno dei punti di frammentazione. Possiamo attribuire la causa di questo fenomeno alla più alta velocità di trasmissione, che implica una capacità meno sensibile alle piccole variazioni di D. L'aumento dei costi computazionali è invece indipendente dal bit-rate e perciò l'andamento decrescente nell'intervallo di padding non è trascurabile.

4 Host collegati tramite uno switch

In questo caso entrambi gli host sono connessi attraverso uno switch che fa uso della tecnica di commutazione "Store and Forward": un pacchetto per essere inviato deve prima essere interamente ricevuto e memorizzato. Poiché i due host non sono collegati in maniera diretta è necessario eseguire il comando per la configurazione del canale su entrambi i computer, in quanto, l'auto-negoziazione avviene con lo switch e non tra di essi.

Quando è presente un solo frammento il RTT vale:

$$\checkmark \quad RTT = 4T_{TX}(D) \quad (3)$$

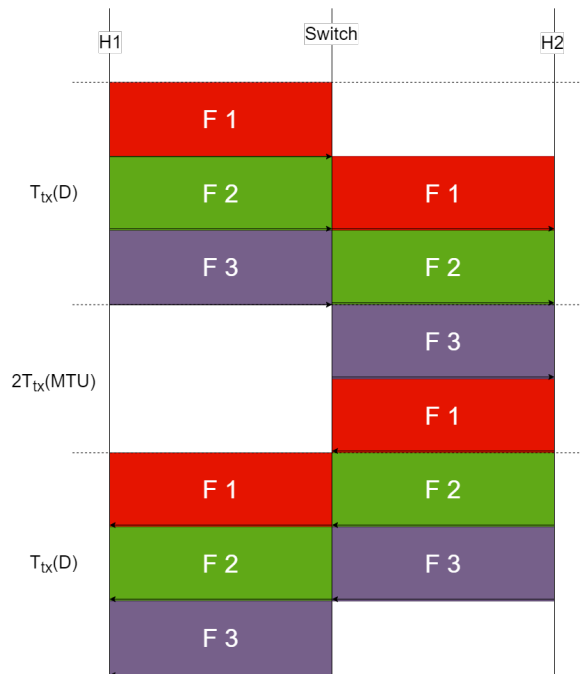


Figura 4.1: Schema di invio pacchetto con switch

Quando i frammenti diventano più d'uno lo switch, che opera soltanto a livello 2, non è in grado di distinguerli e continua ad operare su di essi separatamente come mostrato in figura 4.1. A seguito della frammentazione la formula per il calcolo del RTT diventa quindi:

$$RTT = 2T_{TX}(D) + 2T_{TX}(MTU) \quad \checkmark \quad (4)$$

I tempi di trasmissione del generico dato di dimensione X sono ottenuti come:

$$T_{TX}(X) = \frac{X}{C} \quad (5)$$

Facendo uso delle equazioni mostrate sopra otteniamo la funzione della capacità come:

$$\checkmark \quad \begin{cases} C = \frac{4D}{RTT}, & S < 1473 \\ C = \frac{2D + 2MTU}{RTT} = \frac{2D + 2 * 1538}{RTT}, & s \geq 1473 \end{cases} \quad (6)$$

4.1 Velocità: 10Mb/s

Nel grafico del RTT notiamo che l'andamento è circa lineare e non si discosta di molto da quello ideale. La funzione è definita a tratti come ci aspettiamo dalla formule viste in precedenza. Nel caso del singolo frammento la presenza dello switch ha un effetto negativo (pendenza della retta maggiore) che viene mitigato quando il numero dei frammenti aumenta.

4.2 Velocità: 100Mb/s

In questo caso valgono le considerazioni fatte per il precedente caso senza switch (effetto negativo di η), per quanto riguarda le misure. L'andamento teorico invece è lo stesso del caso precedente a 10 Mbps.

Quale grafico è? Direi da Fig 4.7 a 4.10

Appendice

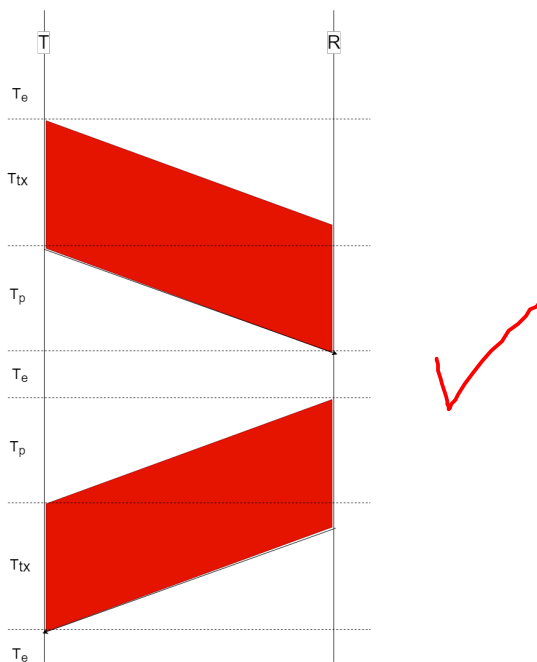


Figura 4.2: Schema di invio pacchetto con collegamento diretto

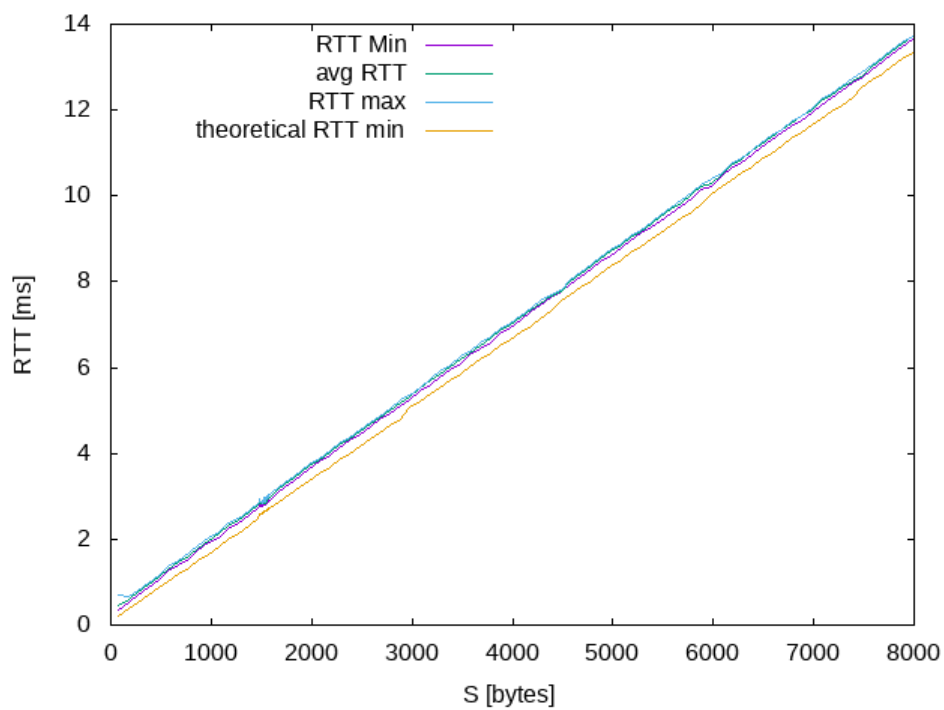


Figura 4.3: Grafico Data-RTT senza switch a 10 Mbps

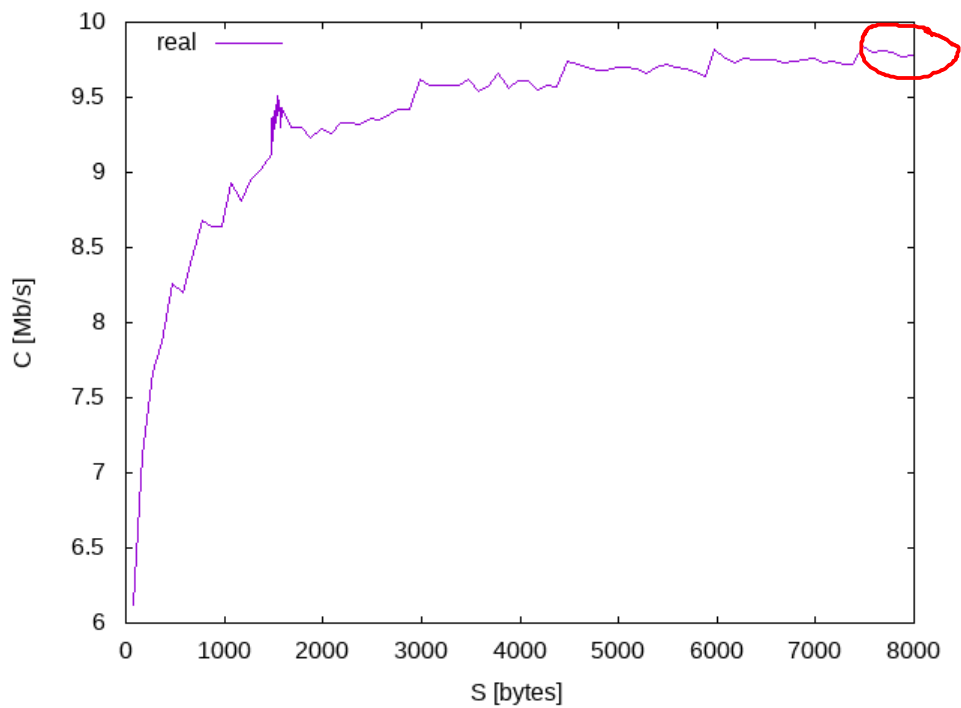


Figura 4.4: Grafico Dati-Capacità senza switch a 10 Mbps

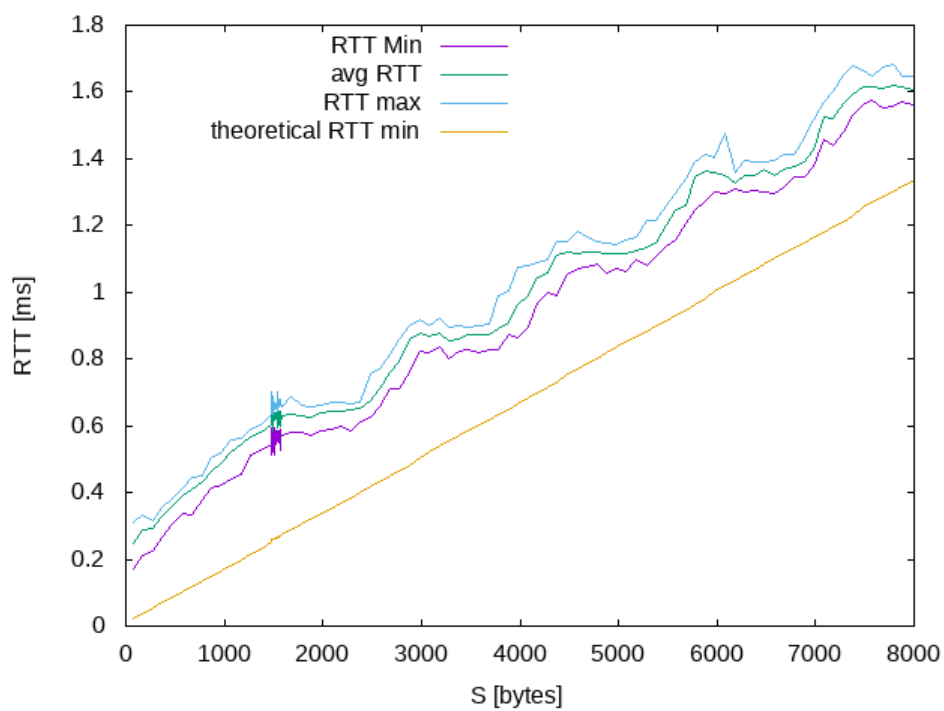


Figura 4.5: Grafico Data-RTT senza switch a 100 Mbps

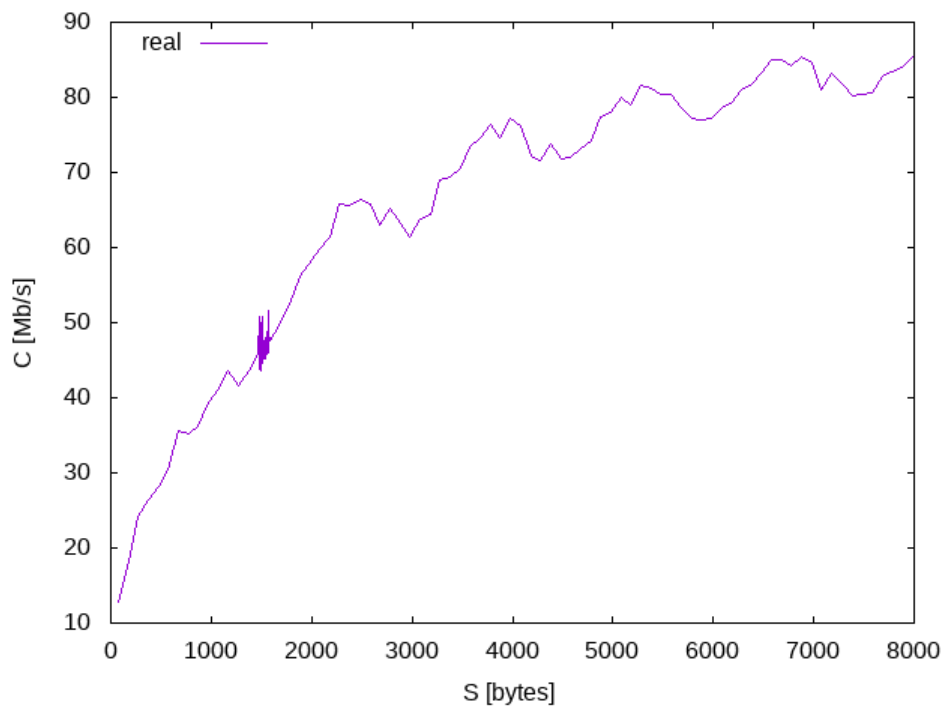


Figura 4.6: Grafico Data-Capacità senza switch a 100 Mbps

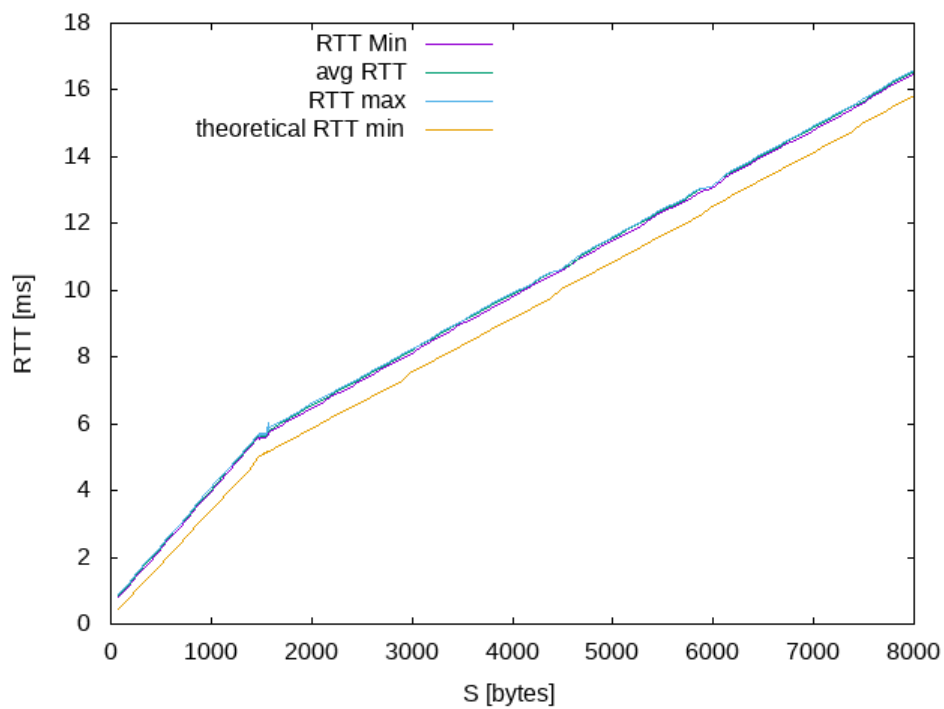


Figura 4.7: Grafico Data-RTT con switch a 10 Mbps

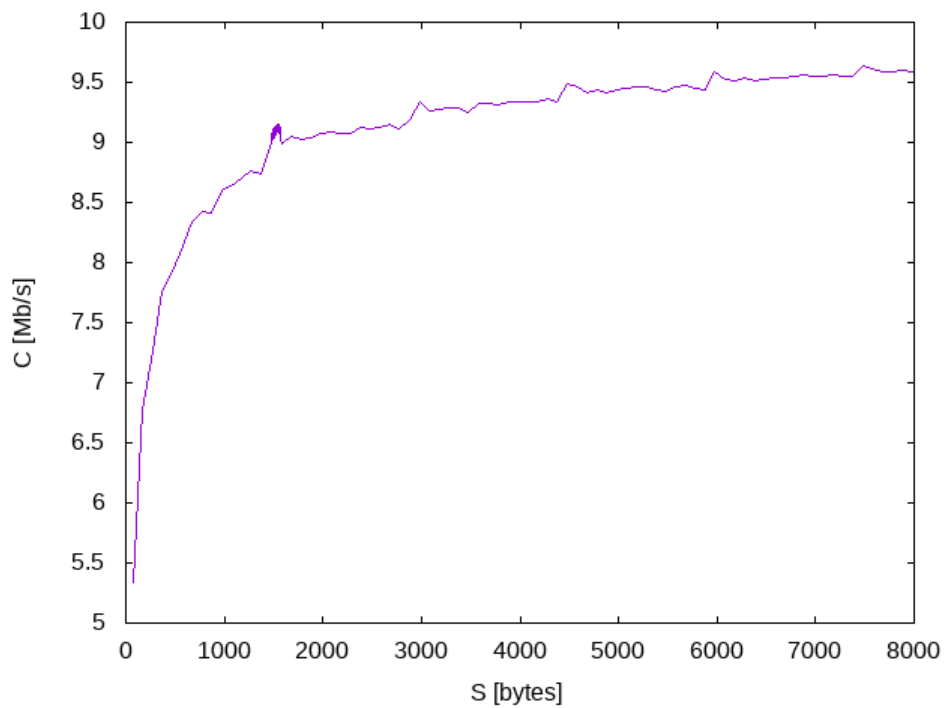


Figura 4.8: Grafico Data-Capacità con switch a 10 Mbps

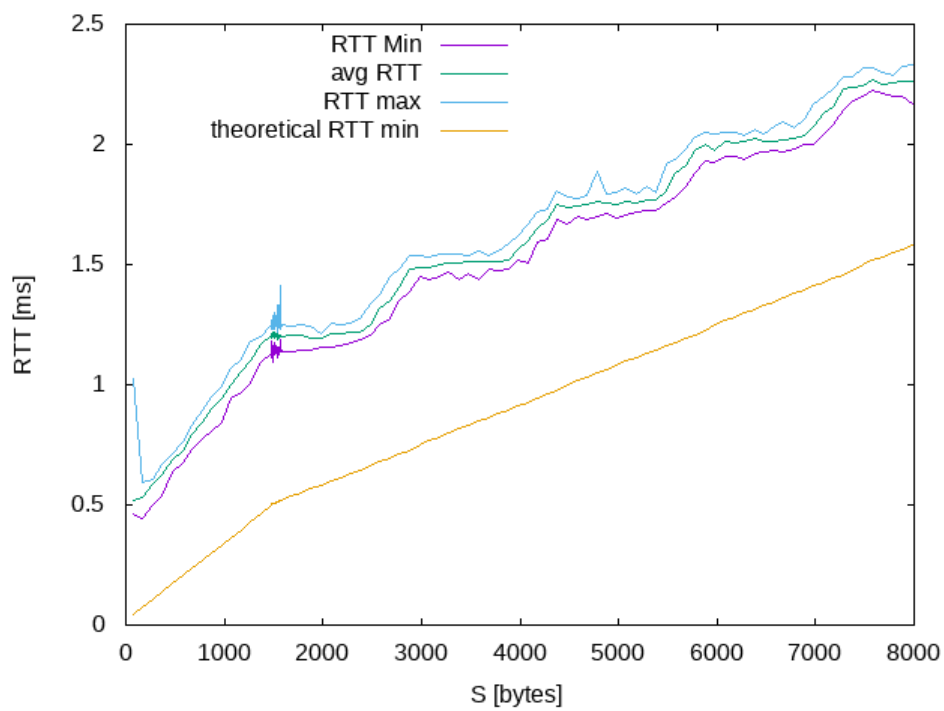


Figura 4.9: Grafico Data-RTT con switch a 100 Mbps

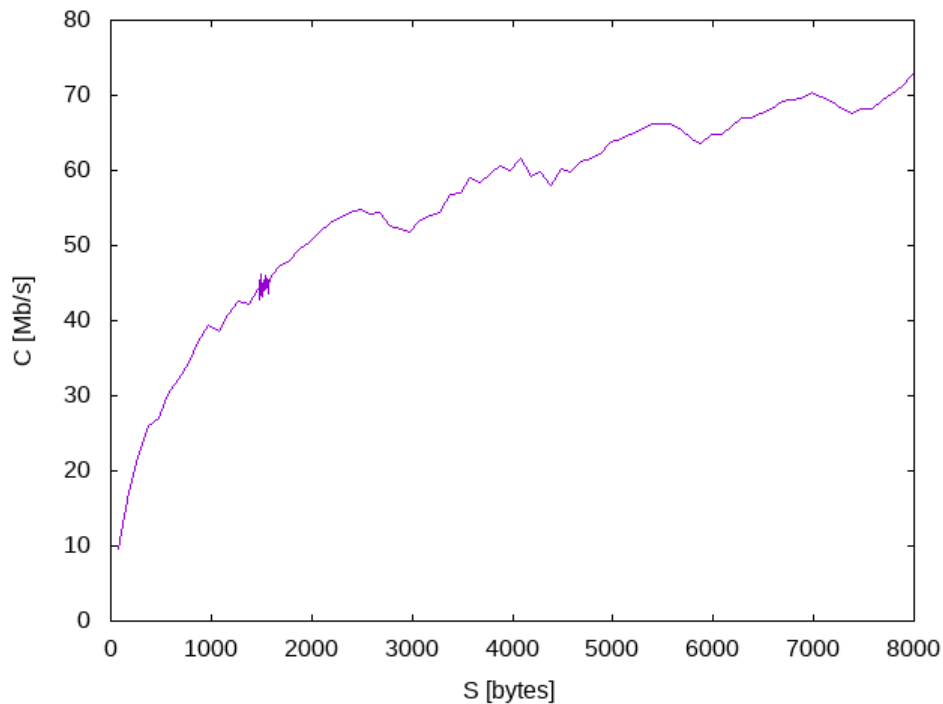


Figura 4.10: Grafico Data-Capacità con switch a 100 Mbps

```

1 # numero di ripetizioni
2 N=20
3 # intervallo tra ripetizioni
4 I=0,02
5
6 # il mio target
7 TARGETS="172.16.14.2"
8
9 # Scegli le lunghezze da usare
10 myLenShort='seq 72 100 1472'
11 myLenFine='seq 1473 1 1573'
12 myLenLong='seq 1580 100 8000'
13
14 #ripeti il ping per ogni lunghezza e per ogni host, estraendo il RTT minimo e la lunghezza
15 for host in $TARGETS; do
16     #cancella i file vecchi per evitare problemi
17     rm -f len${host}.dat rtt${host}.dat all${host}.dat
18     for len in $myLenShort $myLenFine $myLenLong; do
19         echo "pinging host $host with len $len"
20         echo $len >> len${host}.dat
21         sudo ping $host -s $len -c $N -i $I |grep min| cut -d '=' -f 2 | cut -d '/' -f 1,2,3 |tr
22         '/', ' ' >> rtt${host}.dat
23     done
24     #incolla le lunghezze e i rispettivi RTT minimi
25     paste len${host}.dat rtt${host}.dat >> all${host}.dat
26 done

```

Figura 4.11: Script per la raccolta dei dati

```

1 # compute the size at the physical layer of transmitted data
2 # add 1 UDP header of 8 bytes
3 # and k IP+ETH enters for k fragments
4 # the function (ceil((s+8-1)/1480) counts the number of IP fragments.
5
6 D(s)= (s+8)+(20+38)*(1+floor((s+8-1)/1480))
7
8 # compute the eventual padding on the last fragment.
9 padding(x) = ((x)<46)?(46-x):(0)
10
11 # and put everything together.
12 # the size of the last fragment would be ((s+8)%1480 + 20) where the % is the modulus
    operator
13 # that returns the remainder of the division by 1480
14
15 Dtot(s)= D(s) + padding(floor(s+8)%1480+20)
16
17 set term png
18 set out "RTT-10Mbps.png"
19
20 set xlabel "S [bytes]"
21 set ylabel "RTT [ms]"
22 set key left
23 plot 'all172.16.14.2.dat' using ($1):2 title "RTT Min" with lines,\
24      'all172.16.14.2.dat' using ($1):3 title "avg RTT" w lines,\
25      'all172.16.14.2.dat' using ($1):4 title "RTT max" w lines,\
26      'all172.16.14.2.dat' using ($1):(8*2*Dtot($1)/10000.0) title "theoretical RTT min" with
    lines
27 #caso con switch:
28 #      'all172.16.14.2.dat' using ($1):(8*(2*Dtot($1)+2*($1 < 1473 ? Dtot($1):1538))/10000.0)
    title "theoretical RTT min" with lines
29
30 # compute the speed now
31 speed(d,rttMin) = 8*(2*Dtot(d))/rttMin/1000
32 #caso con switch:
33 #speed(d,rttMin) = 8*(2*Dtot(d)+2*($1 < 1473 ? Dtot(d):1538))/rttMin/1000
34
35 set term png
36 set out "speed-10Mbps.png"
37
38 set xlabel "S [bytes]"
39 set ylabel "C [Mb/s]"
40
41 plot 'all172.16.14.2.dat' using ($1):(speed($1, $2)) title "real" with lines

```

Figura 4.12: Script per generazione grafici in gnuplot a 10Mbps (commentato il caso con lo switch)

I 4 esperimenti (10/100 con e senza Switch) sono corretti
e correttamente descritti

10/10