

LABORATORIO DI INTERNET

Report 2: Analisi di velocita' tramite ping

Gruppo 21

Marzo 2021



**POLITECNICO
DI TORINO**

Diego Zanfardino s256536,
Fabio Trovero s258574,
Lorenzo Ferro s260878

prof. Mellia Marco

0 Introduzione

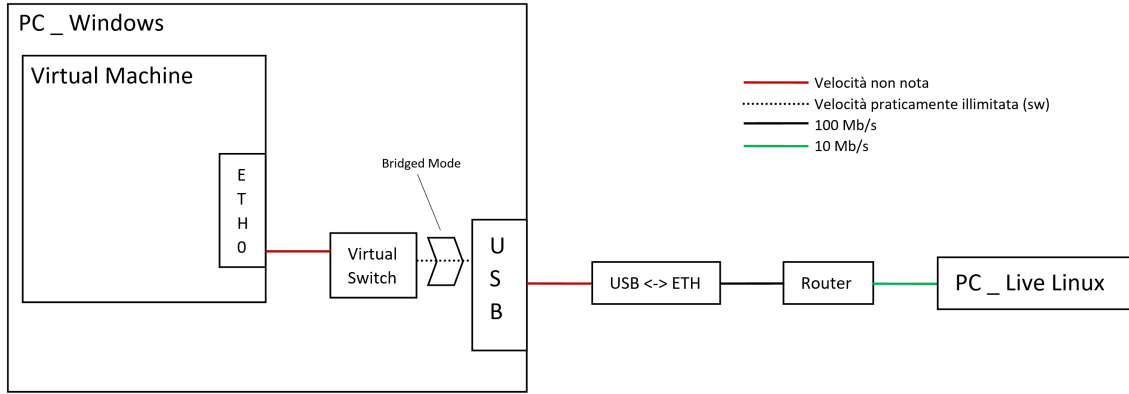
In questo laboratorio analizziamo come poter stimare la velocità di trasmissione a livello fisico tramite misure di RTT, esaminando 4 diverse possibili configurazioni.

Per rendere i grafici più accurati possibile è stato considerato l'effetto dovuto alla frammentazione; quando la dimensione del pacchetto supera la MTU (1500 Byte) calcoliamo il numero di pacchetti risultanti con la seguente formula:

$$D(x) = (x + 8) + (20 + 38) * (1 + \text{floor}(\frac{x + 8 - 1}{1480})) \quad (1)$$

1 Configurazione utilizzata punto 1

In questo primo caso utilizziamo una virtual machine Linux e un pc con una Live USB Linux collegato al primo tramite un router. Abbiamo effettuato l'accesso remoto dal primo al secondo host tramite il comando SSH. Per la connessione si è utilizzato un cavo ethernet (cat 5e). Abbiamo settato dal PC Live Linux tramite il comando "sudo ethtool -s eth0 speed 10 duplex full autoneg on" la velocità di trasmissione con lo switch a 10 Mb/s. Mentre non si può stabilire la reale velocità che abbiamo tra virtual switch e porta ethernet della virtual machine e USB, come si può notare dalla figura che rappresenta la configurazione.



1.1 Risultati

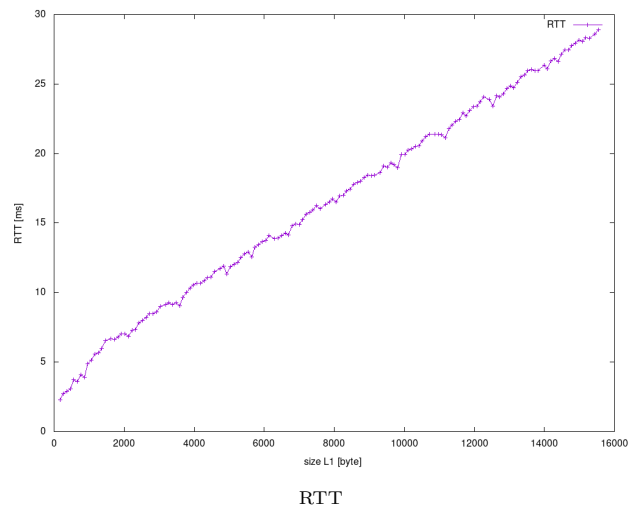
IL grafico riporta l'andamento dell'RTT in funzione dell'aumento della dimensione dei pacchetti. Si può notare una crescita lineare in accordo con quanto ricavato analiticamente, ovvero:

$$\begin{cases} RTT = 4T_{TX} & D < 1500 \\ RTT = 2T_{TX} + 2T_{MTU} & D > 1500 \end{cases} \quad (2)$$

$$T_{MTU} = \frac{1538}{V_{TX}} \quad (3)$$

$$T_{TX} = \frac{D(s)}{V_{TX}} \quad (4)$$

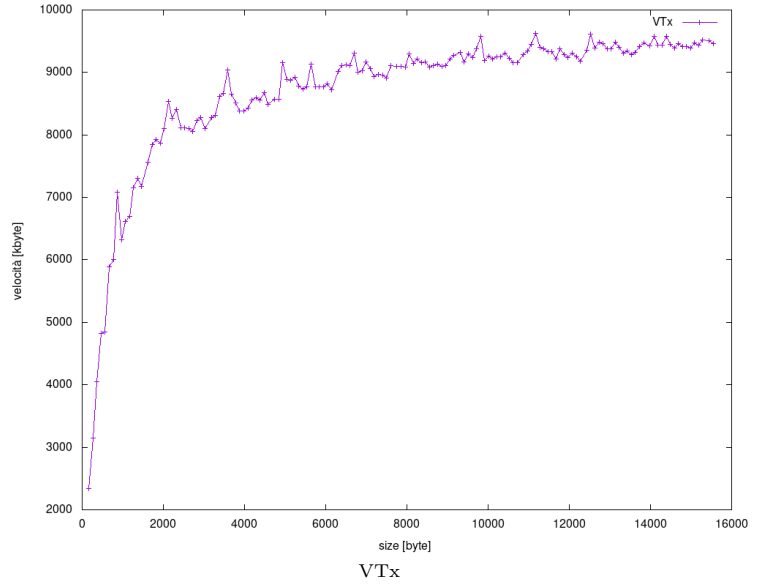
$$RTT = \frac{2D}{V_{TX2}} + \frac{2D}{V_{TX1}} + T_{\eta} \quad (5)$$



Dove V_{TX1} rappresenta la velocità tra USB/ETH e il router, mentre V_{TX2} la velocità tra PC Live Linux e router. T_{η} rappresenta i ritardi propagazione e elaborazione ed è quasi sempre da considerarsi trascurabile. Nel nostro caso è un valore compreso tra 2 e 3 millisecondi poiché equivale al valore in $X=0$ del grafico RTT.

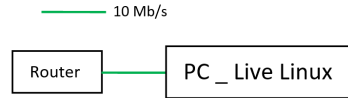
A partire dagli stessi dati e' possibile ricavare l'effettiva velocita' di trasmissione della rete, considerando i contributi della frammentazione come indicato nella formula (2).

Come ci aspettavamo tende asintoticamente a 10Mb/s ossia la velocita' settata inizialmente per la comunicazione tra PC Linux Live e router.

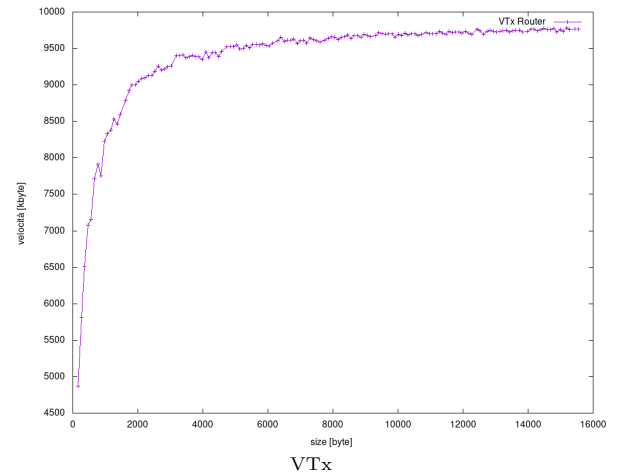
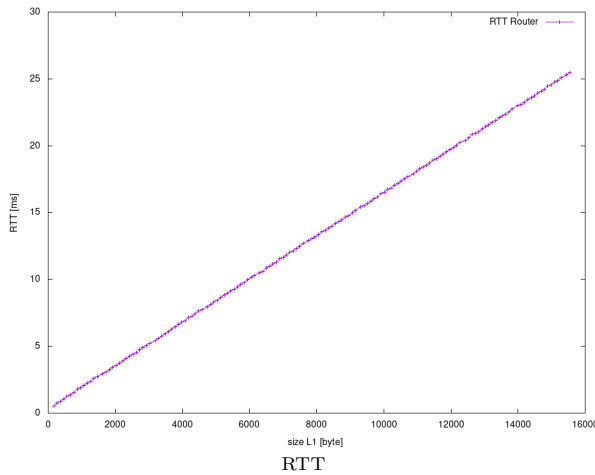


2 Configurazione utilizzata punto 2

In un secondo caso piu' semplice abbiamo coinvolto solo il router e il PC Live Linux. Campionando i primi dati abbiamo notato che in seguito alla prima frammentazione il firewall del router scartava i pacchetti successivi. Abbiamo provato a cambiare la configurazione del firewall dell'interfaccia di rete per permettere la risposta a pacchetti con dimensione maggiore di MTU



2.1 Risultati



In questo caso si rielaborano le formule (2) (4).

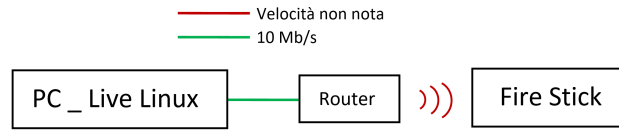
$$RTT = 2T_{TX} + T_{\eta} \quad (6)$$

$$V_{TX} = \frac{2D(s)}{RTT} \quad (7)$$

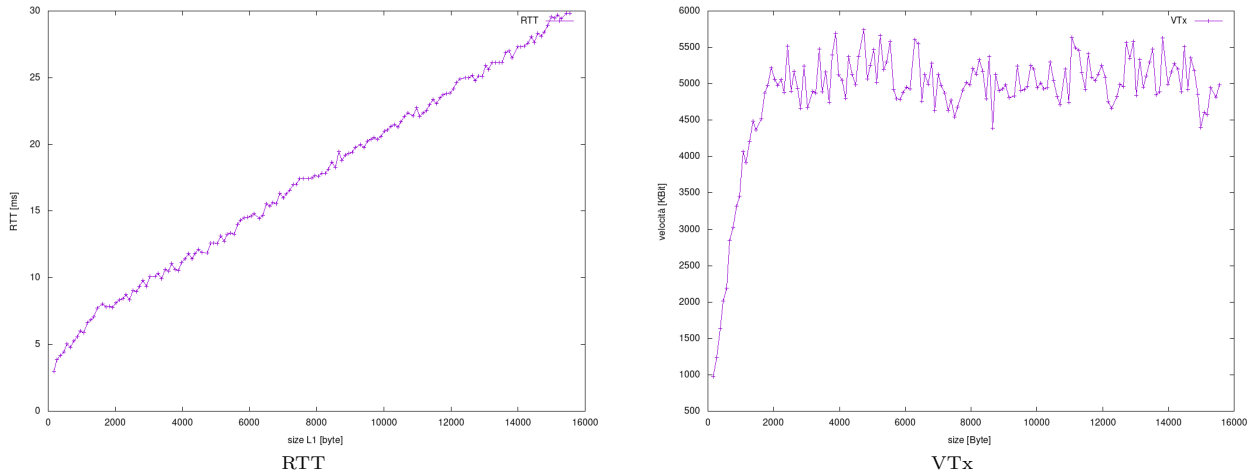
Coinvolgendo solamente due dispositivi collegati mediante un cavo ethernet di modesta lunghezza il comportamtno riscontrato e' soggetto a meno variazioni ed e' molto simile al comportamento teorico.

3 Configurazione utilizzata punto 3

In questo caso si analizza uno scenario in cui dal Pc Live Linux eseguiamo un ping ad un dispositivo connesso via WiFi (nel nostro caso abbiamo utilizzato una firestick TV amazon)



3.1 Risultati



Avendo impostato la velocità di connessione tra Pc Live Linux e il router di 10Mb/s l'unica incognita rimane la velocità della connessione wireless. Ipotizzando il caso in cui $V_{TX1} < V_{TX2}$, e' possibile ricavare:

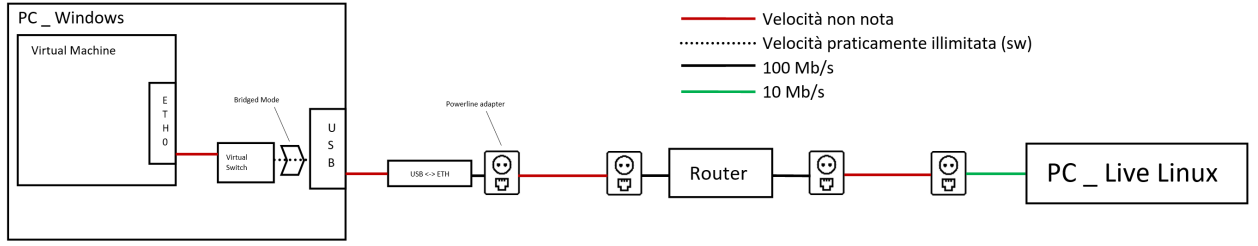
$$\begin{cases} RTT = \frac{2D}{V_{TX2}} + \frac{2D}{V_{TX1}} + T_{\eta} & D < 1500 \\ RTT = \frac{2D}{V_{TX2}} + \frac{2MTU}{V_{TX1}} + T_{\eta} & D > 1500 \end{cases} \quad (8)$$

$$V_{TX1} = \frac{2D}{RTT - \frac{2MTU}{V_{TX2}}} \quad (9)$$

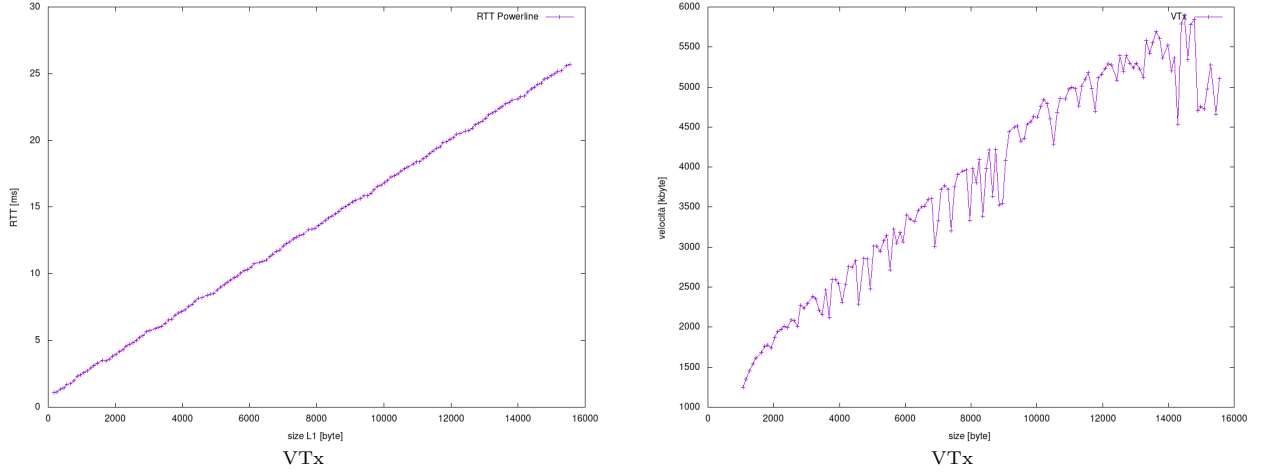
Abbiamo riportato l'andamento della velocità del WiFi approssimando l'header WiFi a 40 Byte. Come si può notare analogamente ai punti precedenti all'aumentare della dimensione del pacchetto si arriva a una stima più accurata della velocità poiché si possono trascurare con meno margine di errore i tempi di propagazione e elaborazione, cioè il fattore T_{η} . Dall'andamento del grafico si può notare un andamento un po' irregolare probabilmente dovuto alla minore affidabilità del mezzo trasmissivo rispetto ad un cavo cablato.

4 Configurazione utilizzata punto 4

In questa configurazione i due host comunicano attraverso adattatori powerline, quindi l'informazione passa attraverso l'impianto elettrico. La velocità di trasmissione così come i protocolli utilizzati a questo livello non sono noti, quindi si ipotizza l'uso del protocollo ethernet e si cerca di stimare la velocità tra i due adattatori powerline. Abbiamo notato utilizzando il comando ethtool che la velocità massima negoziata tra il router e gli host era di 100 Mb/s



4.1 Risultati



Abbiamo ricavato questo modello per approssimare questo scenario ipotizzando tre velocità diverse all'interno del modello, ma una incognita

$$\begin{cases} RTT = \frac{2D}{V_{TX1}} + \frac{4D}{V_{TX2}} + \frac{6D}{V_{TX3}} + T_\eta & D < 1500 \\ RTT = \frac{2D}{V_{TX1}} + \frac{4MTU}{V_{TX2}} + \frac{6D}{V_{TX3}} + T_\eta & D > 1500 \end{cases} \quad (10)$$

Dove:

- V_{TX1} e' la velocità tra Pc Live Linux e il primo adattatore powerline
- V_{TX2} e' la velocità incognita tra gli adattatori powerline
- V_{TX3} e' 100 Mb/s

Dal grafico si può notare un comportamento oscillante e poco stabile probabilmente dovuto a rumore e congestione all'interno della rete elettrica.

A Script per collezionare i dati

```
for((s=100; s<15000; s+=100))
do
    echo -n $s >> dataset.dat
    sudo ping 192.168.1.XX -s $s -c 5 -i 0,05 | grep min
    | cut -d '=' -f2 | cut -d '/' -f1 >> dataset.dat
done
```

B Script configurazione 1

```
set xlabel "size L1 [byte]"
set ylabel "RTT [ms]"

set terminal png size 1024, 768

D(x)=(x+8)+(20+38)*(1+floor((x+8-1)/1480))

set output "RTTp.png"

plot 'rtt_min_h2.dat' using (D($1)):2 title "RTT powerline" with linespoint

set xlabel "size [byte]"
set ylabel "velocità [kbyte]"
set output "VTx.png"

plot 'rtt_min_h2.dat' using (D($1)):(($1<1472 ? (8*4*(D($1))/($2)) :
((8*2*(D($1))/($2)) + (8*2*(D(1538)/($2)))))) :
title "VTx H2 powerline" with linespoint
```

C Script configurazione 2

```
set xlabel "size L1 [byte]"
set ylabel "RTT [ms]"

set terminal png size 1024, 768

D(x)=(x+8)+(20+38)*(1+floor((x+8-1)/1480))

set output "RTTp.png"

plot 'rtt_min_h2.dat' using (D($1)):2 title "RTT powerline" with linespoint

set xlabel "size [byte]"
set ylabel "velocità [kbyte]"
set output "VTx.png"

plot 'rtt_min_h2.dat' using (D($1)):(8*2*(D($1))/($2))
title "VTx H2 powerline" with linespoint
```

D Script configurazione 3

```
set xlabel "size L1 [Byte]"
set ylabel "RTT [ms]"

set terminal png size 1024, 768

D(x)=(x+8)+(20+38)*(1+floor((x+8-1)/1480))

set output "RTTW.png"

plot 'rtt_min_wifi.dat' using (D($1)):2 title "RTT" with linespoint

set xlabel "size [Byte]"
set ylabel "velocità [KBit]"
```

```

set output "VTxW.png"

plot 'rtt_min_wifi.dat' using (D($1)):( D($1)<1500 ?
  (((2*8*10000000*D($1))/( (10000)*($2) - 2*8*D($1) ))*0.001) :
  (((2*8*10000000*1538)/( (10000)*($2) - 2*8*D($1) ))*0.001 ) )
title "VTx" with linespoint

```

E Script configurazione 4

```

set xlabel "size L1 [byte]
set ylabel "RTT [ms]

set terminal png size 1024, 768

D(x)=(x+8)+(20+38)*(1+floor((x+8-1)/1480))

set output "RTTPow.png"

plot 'rtt_min_h2_con_powerline.dat' using (D($1)):2 title "RTT" with linespoint

set xlabel "size [byte]
set ylabel "velocità [kbyte]

set output "VTxPow.png"

plot 'rtt_min_h2_con_powerline.dat' using (D($1)):( D($1)<1500 ?
  (4*8*D($1)*10**7*10**8/( ($2)*10**4*10**8 - (2*8*10**8*D($1)) -
  (6*8*10**7*D($1)) ))*0.001 : (4*8*1538*10**7*10**8/( ($2)*10**4*10**8 -
  (2*8*10**8*D($1)) - (6*8*10**7*D($1)) ))*0.001 ) with linespoint

```