

LAB PING PONG

Malchiodi Riccardo	S5680500
Giulia Ponassi	S6145273
Andrei Dorin Sandu	S6483324

Settimana 1

tcp_ping

Per compilare TCP dobbiamo connettere il nostro client ping ad un server pong. Possiamo usare gli eseguibili presenti già nella cartella bin ossia i **gcUbuntu64_pong_server** o alternativamente (se è aperto) seti.dibris.unige.it (Ipv4: 130.251.61.30) nella **porta 1491**.

Per debuggare in localhost, usiamo due terminali: nel primo eseguiamo il pong server fornito scrivendo **./gcUbuntu64_pong_server 1491**, mentre nel secondo eseguiamo il nostro tcp ping scrivendo **./tcp_ping localhost 1491 64**.

Per debuggare usando il server pong di seti.dibris.unige.it usiamo un solo terminale dove scriviamo **./tcp_ping 130.251.61.30 1491 64**.

Settimana 2

udp_ping.c

Il procedimento per compilare è analogo al TCP ma ovviamente usando il codice UDP.

Settimana 3

Pong_server.c

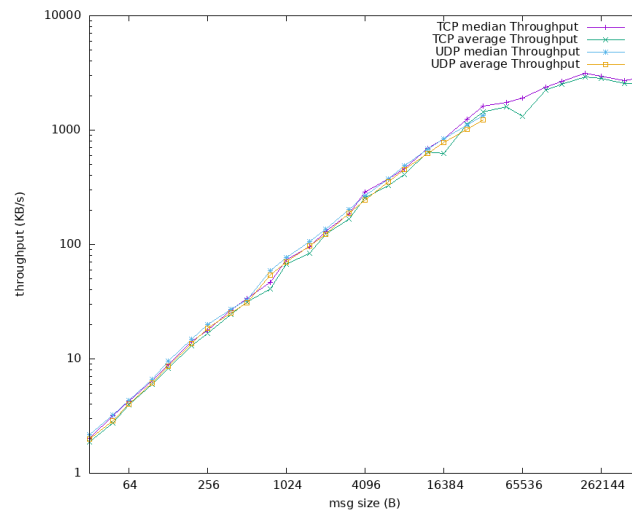
Per prima cosa andiamo sulla cartella scripts, scriviamo **chmod u+x mkfile.bash**, e lo eseguiamo facendo **./mkfile.bash**.

Dopo aver lanciato un certo numero di interazioni Ping/Pong TCP e UDP, nella directory data sono creati dei file che contengono i risultati delle interazioni stesse (es. tcp_32.out, udp_64.out, ecc.). Se i file non si creano automaticamente, possiamo fare make nella cartella datas. Dopo aver creato questi file verranno creati in automatico i file **udp_throughput.dat** e **tcp_throughput.dat**.

Andiamo successivamente, sempre nella cartella scripts, a scrivere **chmod u+x gplot.bash** e lo eseguiamo facendo **./gplot.bash** (prima si deve scaricare gplot).

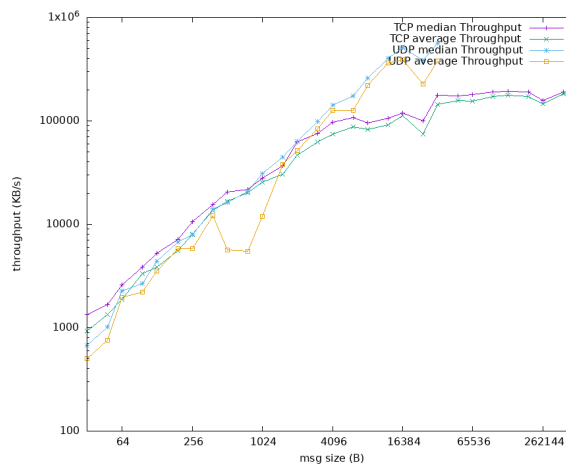
Output ottenuti

Throughput ottenuto con server dibris:



- Il throughput è limitato dalla rete esterna, quindi i valori massimi sono moderati.
- Si vede una crescita graduale del throughput, ma stabilisce un limite massimo che è più basso rispetto a una rete locale.

Throughput ottenuto con pong server (in localhost) implementato da noi :



- I valori di throughput sono molto più alti rispetto a quelli del throughput misurato su `seti.dibris.unige.it`, soprattutto per i messaggi grandi.

Settimana 4

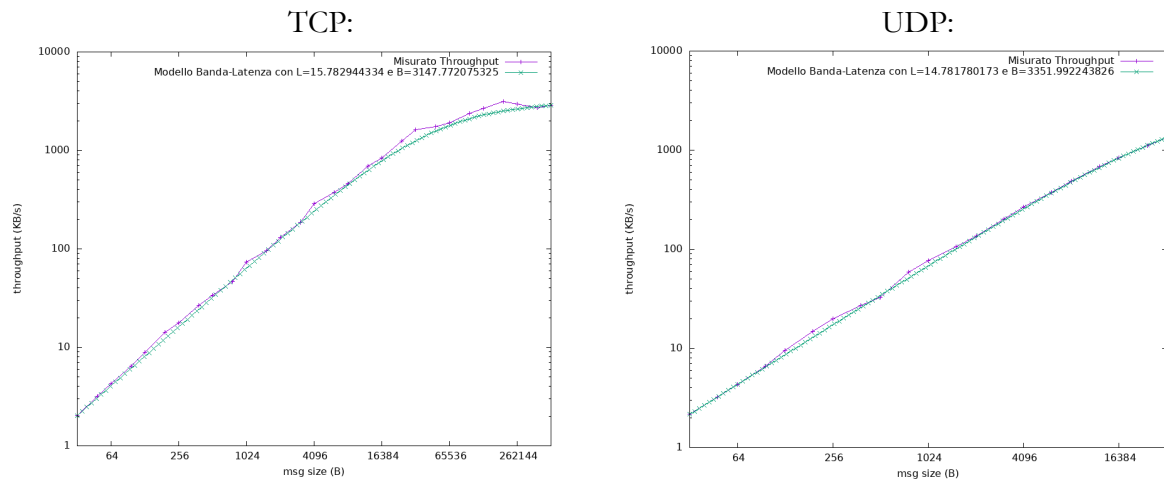
latencyBandModel.bash

Dobbiamo Scaricare bc per poter compilare. Facciamo `chmod u+x latencybandmodel.bash`, poi lo eseguiamo con `./latencybashmodel.bash`

Per ottenere i 2 grafici di tcp e udp dobbiamo scrivere `./latencybashmodel.bash tcp` e `./latencybashmodel.bash udp`.

Output Ottenuti

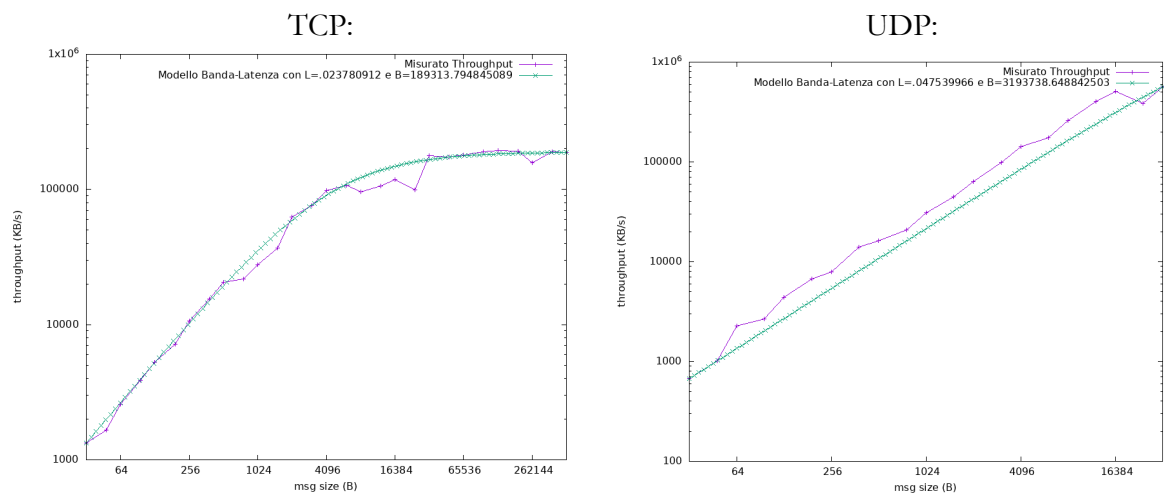
Nel nostro caso, usando il server pong di `seti.dibris.unige.it`, otteniamo:



Nel TCP il valore della latenza è alto perché stiamo comunicando su una rete esterna tra client e server (`seti.dibris.unige.it`), e quindi ci sono ritardi dovuti alla distanza e ai vari passaggi nella rete. Il valore di banda è ridotto, visto che la banda disponibile è limitata dalla rete esterna.

Nell'UDP il valore della latenza è alto, simile al caso TCP, il che indica che anche per UDP c'è un ritardo significativo quando si comunica su una rete esterna. Il valore della banda, invece, è leggermente più alto rispetto a TCP, il che significa che UDP riesce a sfruttarla di più. UDP non gestisce il controllo della connessione come TCP, quindi può ottenere una banda leggermente maggiore, compromettendo però la stabilità

Mentre con il server pong implementato da noi otteniamo:



Il valore della latenza è molto basso, dovuto al fatto che client e server sono sulla stessa macchina. Il valore di B è molto più alto rispetto ai grafici ottenuti tramite `seti.dibris.unige.it`, perché in localhost il throughput non è limitato da una rete esterna ma solo dalle capacità della macchina stessa.

Il throughput segue il modello, e a volte risulta più alto: l'assenza di limitazioni fisiche di banda in localhost, fa sì che il throughput possa crescere oltre il limite teorico impostato dal modello Banda-Latenza.