

Relazione esercitazione “Pingpong”

Stefano Casassa (6421009), Matteo Tondo (6434023), Gianluca Biffi (6366900)

Scopo:

L'obiettivo dell'esercitazione è di sviluppare, completando delle porzioni di codice omesse, la simulazione di una comunicazione tra Client e Server, utilizzando i due principali protocolli del livello di trasporto: UDP e TCP, e di effettuare un'analisi delle “performance” in relazione alla grandezza dei pacchetti durante la comunicazione dei due protocolli.

Prima parte TCP_ping

Nella prima fase il compito era di completare i file relativi al ping TCP e alla telemetria. Dopo aver consultato la documentazione presentata, non abbiamo riscontrato problemi nell'implementazione del socket TCP e nello sviluppo della funzione `timespec_delta()`.

Seconda parte UDP_ping

Nella seconda parte si doveva completare la sezione relativa al protocollo UDP.

Nell'implementazione abbiamo riscontrato diverse problematiche, al punto che siamo arrivati a completarla dopo la terza parte.

Abbiamo utilizzato il tool `gdb` per effettuare il debugging, il problema riscontrato era il seguente: La connessione veniva effettuata, sia in locale che con il server del dibris, ma subito dopo il programma si bloccava, restituendo in output l'errore “Error recieve: Resource temporarily unavailable”.

Durante il debugging con `gdb` però, se eseguito completamente con il comando `next`, l'esecuzione avveniva correttamente.

Dopo ricerche online siamo giunti alla conclusione che si trattasse di un problema con la variabile `recv_byte`, in quanto erroneamente controllavamo eventuali perdite di byte della funzione `recv`.

Ciò è normale che accada con il protocollo UDP, quindi rimuovendo il controllo abbiamo risolto il problema.

Abbiamo supposto che il motivo per il quale non abbiamo riscontrato il problema con `gdb` è perché abbiamo ridotto il numero di ripetizioni (dal file `pingpong.h`) per velocizzare il debugging e questo diminuiva la possibilità di perdite di byte.

Un altro motivo per cui abbiamo avuto così tante difficoltà è che credevamo che il processo si bloccasse all'inizio, siccome non avevamo nessun output a terminale, questo avveniva (supponiamo) siccome le stampe di output sono state messe in coda, dando precedenza alla comunicazione.

Terza parte Pong

Nel completare la terza parte non abbiamo riscontrato alcun problema.

Avviando il server pong con il client TCP sulla stessa macchina siamo riusciti a verificarne il funzionamento corretto.

Abbiamo avuto qualche problema con la parte relativa al client UDP, tuttavia, come specificato sovrastante, siamo giunti alla conclusione che il problema proveniva dal file `UDP_ping`.

Quarta parte

Nella fase finale del progetto, abbiamo completato il file bash `latencyBandModel` per generare i grafici relativi al Throughput dei protocolli UDP e TCP.

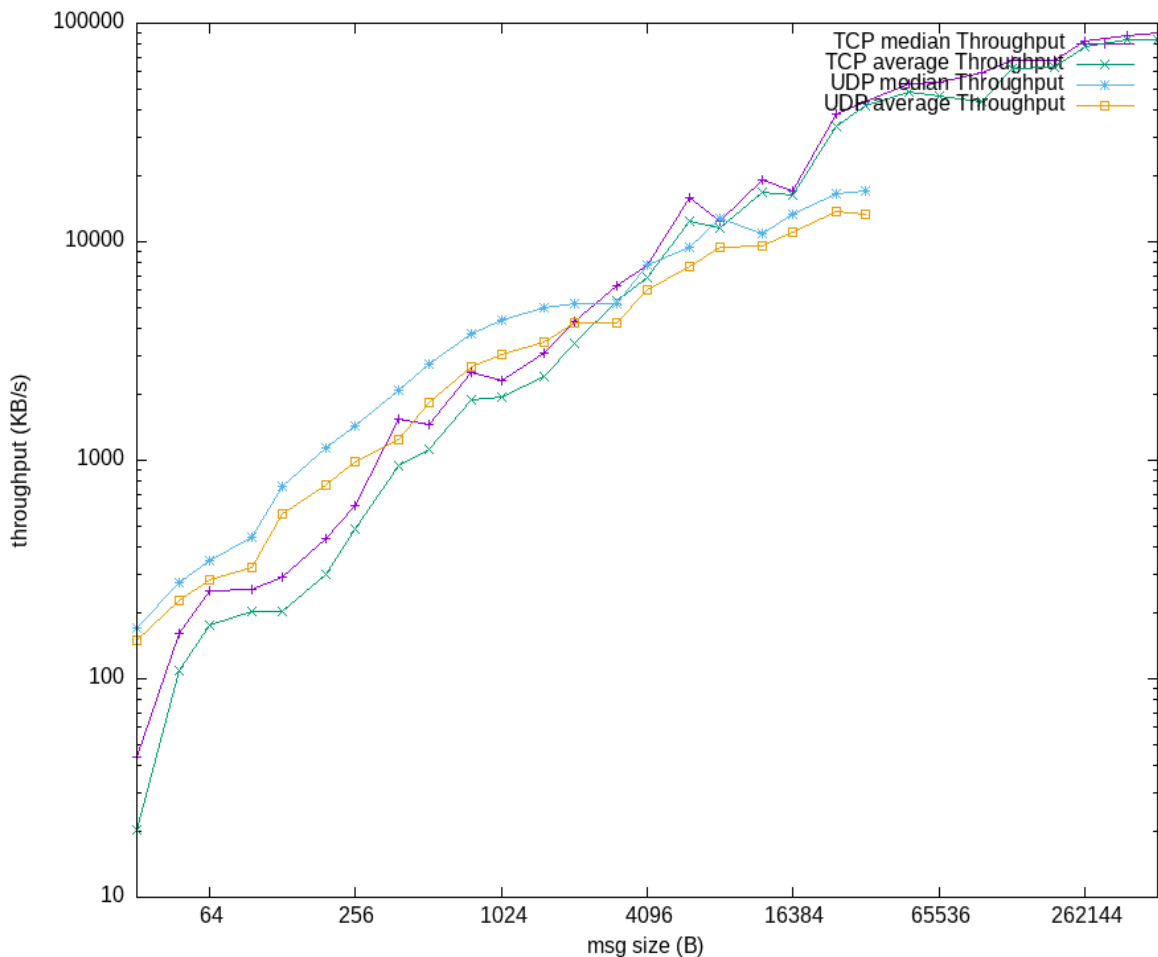
Non abbiamo riscontrato particolari difficoltà nella parte di ricavo dei parametri.

Per fare ciò abbiamo utilizzato un file .c di supporto, chiamato BeLCalc, per calcolare il valore di banda e latenza. Passando come argomenti a terminale i valori di N1, T(N1), N2, T(N2), ed un valore compreso tra 1 e 2 (1 = Banda, 2 = Latenza).

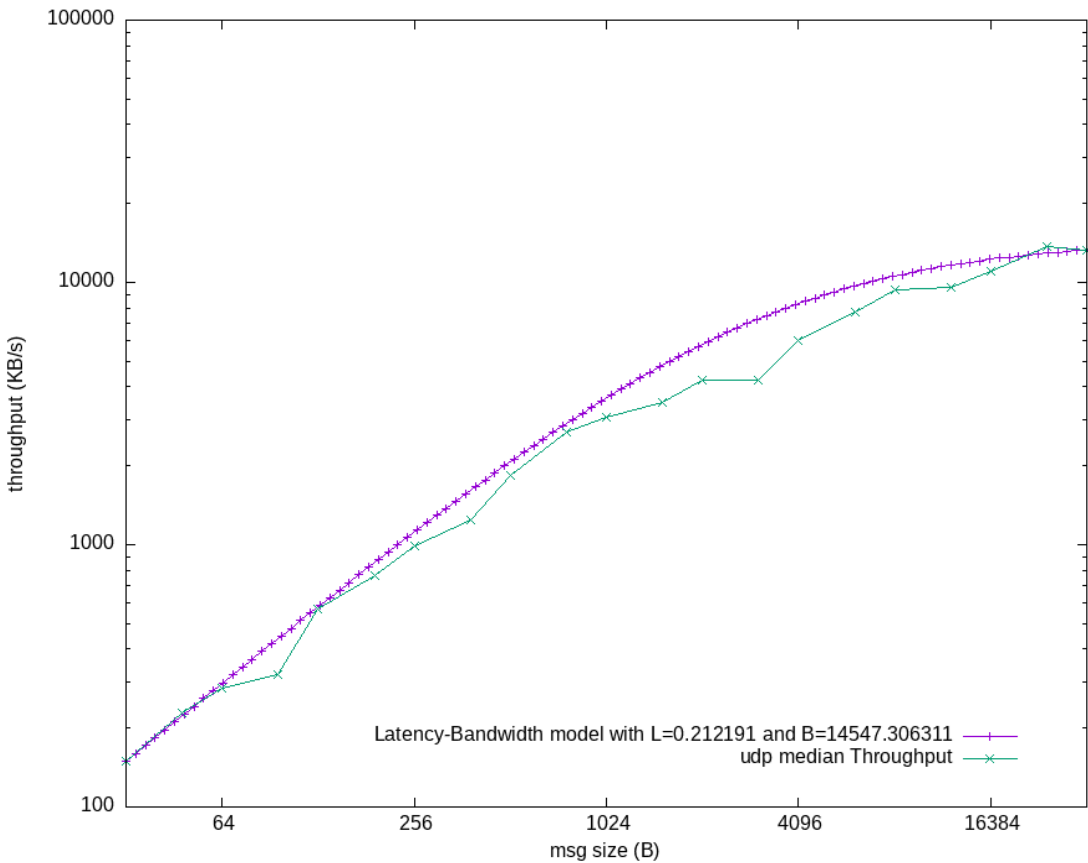
Nella realizzazione dei grafici abbiamo riscontrato un errore di coerenza nel grafico, in quanto nel file .c di supporto abbiamo sbagliato l'operazione del calcolo della latenza, successivamente corretta.

Grafici generati:

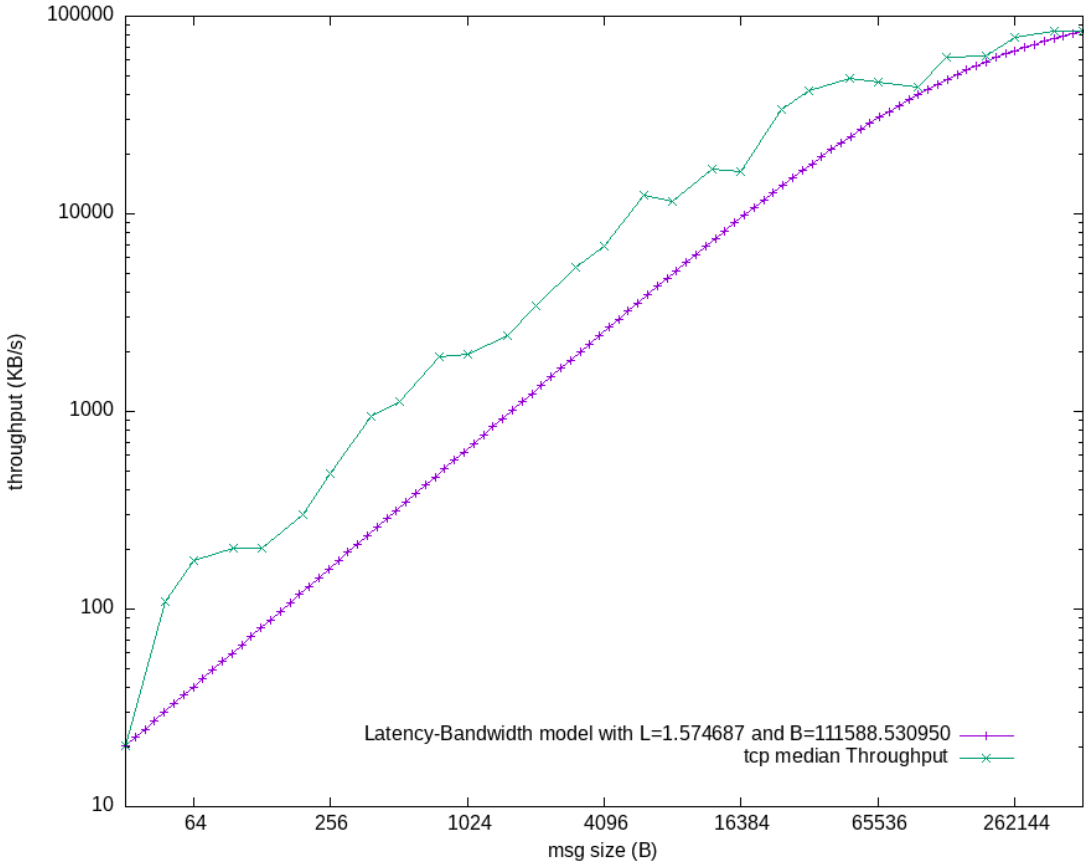
Throughput:



UDP Throughput:



TCP Throughput:



In entrambi i grafici in viola è rappresentata la curva teorica, mentre in verde quella effettiva. Com'è possibile verificare dai grafici il modello teorico differisce dalla realtà, ma i risultati sono coerenti.

Nella misurazione dell'UDP Throughput possiamo vedere come la curva teorica non si distacchi di molto dalla "curva" effettiva; mentre nel caso del TCP Throughput siamo stati più sfortunati, siccome il throughput è stato maggiore di quello teorico per gran parte delle misurazioni. Inoltre, succede proprio nell'ipotesi dello spunto di riflessione, ovvero l'ultima riga non corrisponde al valore più grande di throughput.

I test sono stati effettuati utilizzando i computer di SW1, connettendoci al server pong del DIBRIS (seti.dibris.unige.it : 130.251.61.30, porta 1491).

Spunto di riflessione: cosa succede se, a causa di errori di misura, l'ultima riga non corrisponde al valore più grande di throughput misurato? Come si potrebbe correggere la situazione?

In tal caso, nel grafico noteremo che il punto massimo raggiunto dalla curva teorica sarà inferiore a quello del throughput effettivo, nonostante inizio e fine continueranno a coincidere.

Una possibile soluzione potrebbe essere prendere come valore massimo della curva teorica il valore effettivo con throughput massimo. In questo caso però l'estremo destro delle due curve non coinciderebbe.

Un'ulteriore soluzione, potrebbe essere quella di prendere più valori di riferimento per calcolare la curva teorica, rendendola più precisa.