

# Laboratorio 1: PING-PONG

## Obiettivi:

Lo scopo del laboratorio era quello di implementare programmi che usassero due tipi diversi di comunicazione: TCP e UDP, per poi confrontarne le velocità.

## Macchina utilizzata:

Per effettuare la fase di testing del programma è stato utilizzato un computer con processore Intel, sfruttando il sottosistema Linux per Windows.

## Come funzionano i programmi:

Il primo programma, che utilizza socket TCP, tenta di connettersi ad un server per inviare messaggi di una dimensione stabilita dall'utente.

Il server, dopo aver ricevuto il messaggio, invierà un messaggio di risposta di medesime dimensioni, in modo da poter permettere al client, salvando il tempo di invio e ricezione del messaggio, di calcolare il RTT.

Per avere un RTT corretto bisogna però sottrarre il tempo impiegato dal server per elaborare la richiesta: infatti è il server stesso a salvare le tempistiche per inviarle all'interno della risposta.

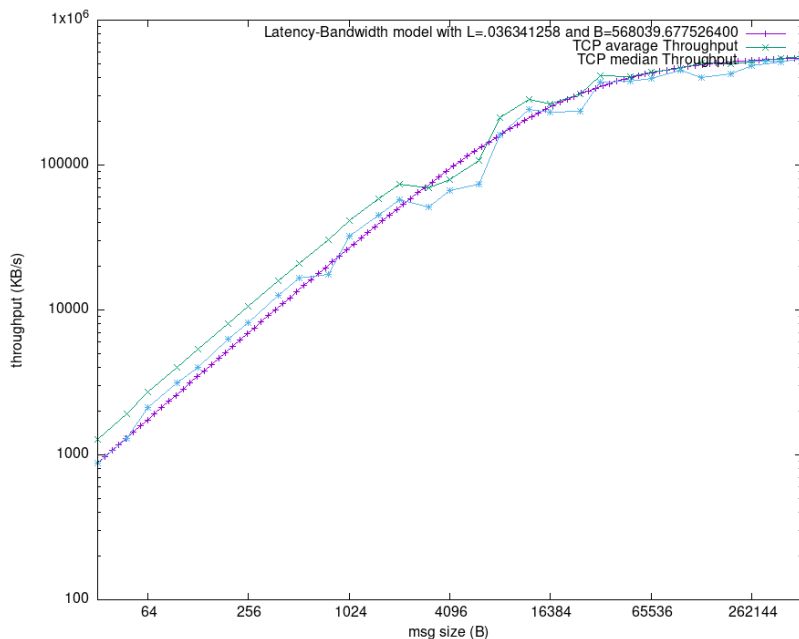
Troviamo un meccanismo analogo anche utilizzando socket di tipo UDP.

## Come abbiamo lavorato:

Inizialmente sono stati preparati i programmi di ping e testati grazie all'eseguibile fornito. Una volta assicurata la correttezza di essi, ci siamo potuti dedicare allo sviluppo del server, fino ad avere entrambe le parti funzionanti.

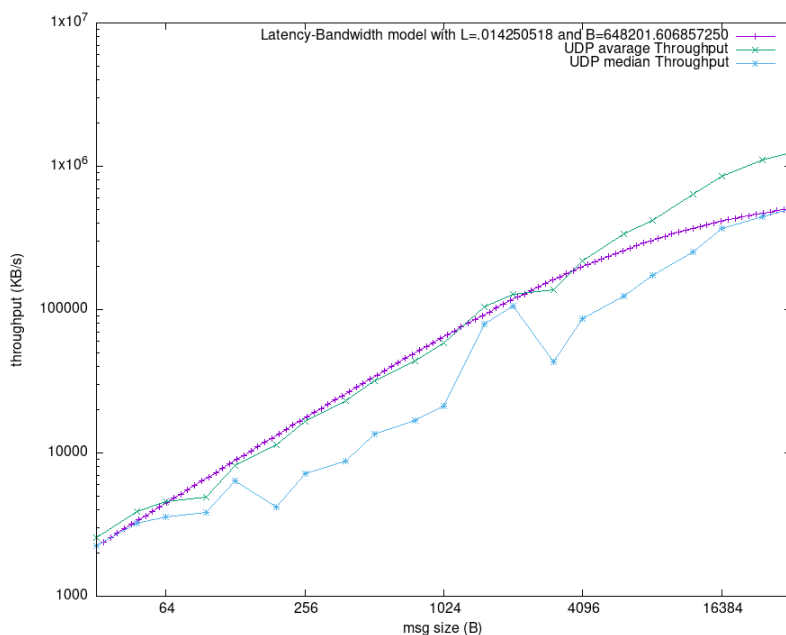
Non avendo familiarità con il linguaggio bash, non siamo riusciti a implementare lo script per lo sviluppo di grafici per tempo. Durante il semestre siamo però riusciti a comprenderne meglio il funzionamento, per poter poi implementare lo script che ci ha permesso di ottenere i grafici che seguono.

## Analisi dei grafici:



In questo primo grafico analizziamo le prestazioni del protocollo TCP. La linea verde (**media**) e quella azzurra (**mediana**) ci permettono di notare come l'aumento di dati inviati sia direttamente proporzionale all'aumento dell'efficienza. Questa curva tenderà ad appiattirsi una volta raggiunta la capacità di rete.

La funzione  $f(x) = L + \frac{x}{B}$  ci permette di avere una stima del **throughput** e di comprendere l'efficacia del protocollo.



Questo grafico mostra la medesima analisi del primo, ma sul protocollo UDP. Possiamo notare come sia stato particolarmente efficiente, nonostante sia un modello di comunicazione piuttosto inaffidabile, tranne quando non vengono persi pacchetti durante la comunicazione e, avendo eseguito i test su localhost, è stato il nostro caso.

Abbiamo potuto quindi osservare come il protocollo UDP risulti mediamente più efficiente, ma solo grazie alle condizioni in cui sono stati svolti i test. Il TCP, in

condizioni meno "perfette", sarebbe sicuramente risultato più affidabile rispetto al predecessore che non presenta un sistema di conferma per la ricezione dei pacchetti.

### Analisi casi particolari:

**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?

Dato che il throughput viene calcolato utilizzando il primo e l'ultimo valore, avremmo come risultato della funzione un valore inaffidabile ed estremamente più piccolo. Si potrebbe gestire questo problema analizzando gli altri valori presenti del .dat, confrontandoli per assicurarsi di ottenere il valore più grande.