Primo laboratorio – Ping Pong

Federica Tamerisco - William Chen - Marco Chen

In questo laboratorio abbiamo implementato due processi client Ping (TCP e UDP) che si connettono a un server remoto Pong con porta >= 1491.

Nel caso del TCP, il Client manda una richiesta di connessione al server, già in ascolto sulla porta, inviando una stringa di dati.

Il Pong server riceve la richiesta, controllando la corrispondenza dei parametri, e risponde al client.

In caso di errore, esso viene comunicato al client e chiude la comunicazione.

Se il controllo è andato a buon fine, il server si mette in esecuzione, seguendo le indicazioni all’interno del messaggio del client.

Dopodiché, il client invia il primo messaggio contenente i dati e il server glielo restituisce nel minor tempo possibile.

Infine, il client calcola l’RTT (Round Trip Time), per permettere al messaggio di compiere il percorso di “andata e ritorno”, e completa il numero di ripetizioni concordate.

Nel caso dell’UDP, esso si connette in ugual modo al server, utilizzando la connessione del TCP; quindi il client manda una stringa di dati per richiedere la connessione.

Il server riceve il messaggio e compie i controlli come descritto sopra.

Nel caso in cui i controlli vengano superati, il client chiude la connessione TCP e crea il socket di tipo DGRAM; a questo punto si inizia l’invio di DATAGRAM UDP al server, il quale lo restituisce nel minor tempo possibile.

Il client quindi calcola l’RTT e si procede con lo stesso metodo di prima.

Il server quindi si occupa di gestire la connessione con almeno un client tramite l’utilizzo passivo di socket e relative system call.

Una volta ricevuta la richiesta di connessione, il server si duplica e si occupa del client connesso al figlio appena generato e al termine dell’esecuzione, il processo termina; il padre attende nuove connessioni.

Oltre a client e server, abbiamo implementato anche uno script bash che calcola i valori di banda e latenza per stimare le prestazioni della rete e produrre dei grafici per ognuno dei protocolli.

Lo script che si viene a creare è “*bandwidth\_latency.bash*” e contiene:

* Banda: tasso di trasferimento dei dati in un certo intervallo di tempo. All’aumentare della banda aumenta il flusso di dati.
* Latenza: tempo impiegato per far viaggiare i dati nella rete, da un punto all’altro.

Il modello bandwidth-latency è utile nella stima del throughput (indicato con T), cioè la quantità di dati trasmessi in un certo intervallo di tempo partendo da un numero limitato di misurazione.

Possiamo riassumere tutto con queste 2 formule:

Dove Nmin e Nmax sono il valori minimo/massimo di dimensione dei messaggi inviati dai client, mentre il Delay è il tempo medio per arrivare a destinazione e si calcola:

Debugging e Testing

Entrambi sono stati eseguiti su WSL Ubuntu 22.04 LTS su macchina Win11.

Grafici

Commenti

Problemi irrisolti

Il problema principale che abbiamo riscontrato con questo laboratorio è il fatto che la produzione dei grafici non va a buon fine e se ne ottiene solo uno, cioè quello del throughput.

Inoltre, usando il nostro pong\_server riscontriamo l’errore “UDP Pong failed sending datagram back: Bad file descriptor”.

Solo eseguendo il gc\_pong\_server riusciamo a ovviare a questo problema.