

Relazione pingpong

Componenti:

- Samuele Barrago
- Daniele Sacco
- Lorenzo Livio Vaccarecci

Il Pong Server è un'applicazione che risponde ai messaggi inviati dai client Ping, sia tramite protocollo TCP che UDP. Il server è progettato per accettare connessioni multiple da client Ping e rispondere alle richieste in modo efficiente.

Errori e Soluzioni

Durante lo sviluppo del Pong Server, abbiamo riscontrato e risolto diversi errori. Di seguito sono riportati alcuni esempi di come abbiamo modificato il codice per risolvere questi problemi.

- Invio dei pacchetti
 - Problema: L'invio di messaggi attraverso il socket in modalità bloccante non riusciva a inviare tutti i dati.
 - Soluzione: Abbiamo sostituito la funzione 'send' con la nostra funzione personalizzata 'blocking_write_all'.
- Scrittura msg_no all'inizio del buffer
 - Problema: La scrittura del numero del messaggio all'inizio del buffer non funzionava correttamente
 - Soluzione: Abbiamo utilizzato la funzione 'sprintf' per formattare correttamente la stringa.
- Ricevimento pacchetti dal socket
 - Problema: La ricezione di risposte attraverso il socket in modalità non bloccante con timeout presentava problemi.
 - Soluzione: Abbiamo implementato un ciclo for per ricevere i dati in più passaggi, invece di cercare di ricevere tutti i dati in una sola volta.
- Segmentation fault alla ricezione di tutti i pacchetti
 - Problema: una volta che il socket riceveva tutte le risposte, il programma sollevava un segmentation fault.
 - Soluzione: il problema riguardava come trattavamo gli argomenti della funzione 'timespec_delta2milliseconds'.

Debug

Durante lo sviluppo del server Pong, abbiamo adottato un approccio passo-passo per il debug. Inizialmente, abbiamo testato il nostro server con 'gc_*ping', poi con i nostri ping. Abbiamo utilizzato la modalità debug del Makefile per individuare e risolvere gli errori. Questo ci ha permesso di migliorare il nostro server in modo efficace.

Utilizzo

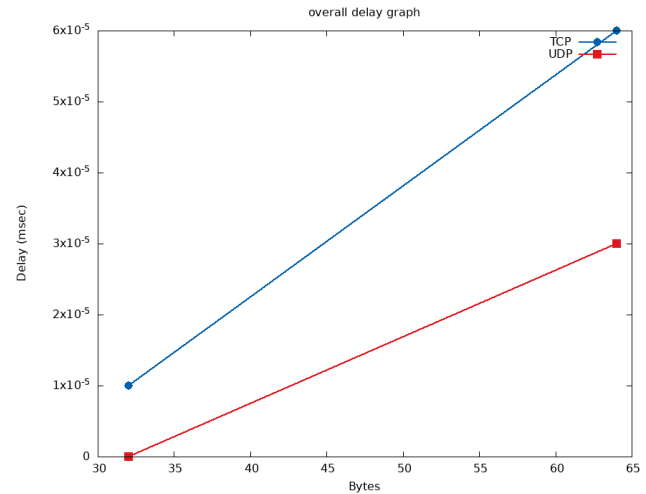
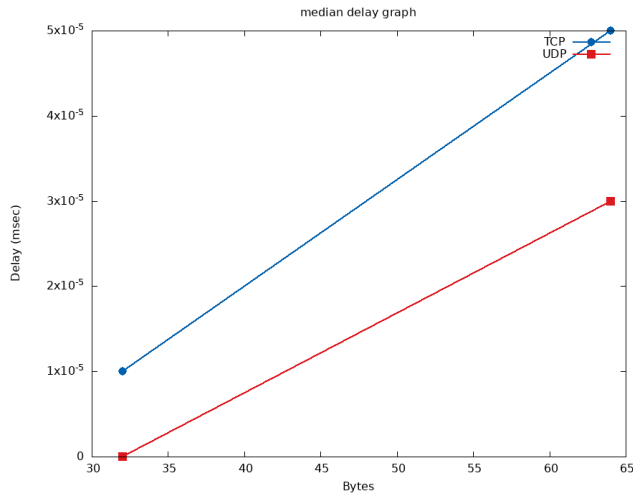
Il Pong Server deve essere eseguito con un argomento che specifica la porta su cui ascoltare. Ad esempio:

```
./pong_server 1491
```

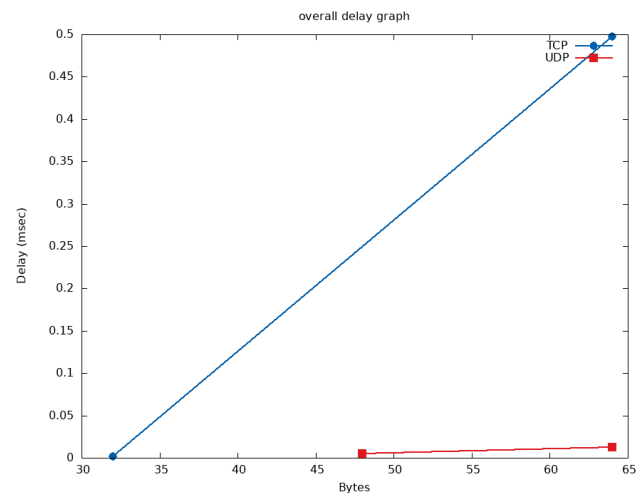
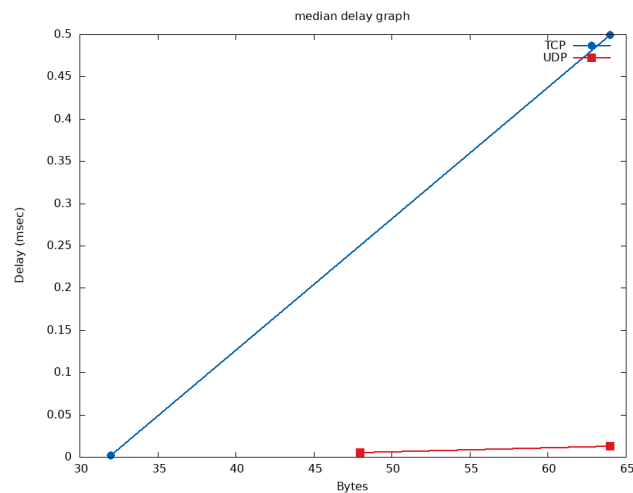
Il server rimarrà in ascolto su questa porta, gestirà le connessioni dei client Ping e risponderà alle loro richieste.

Grafici del delay in funzione dei bytes ricevuti

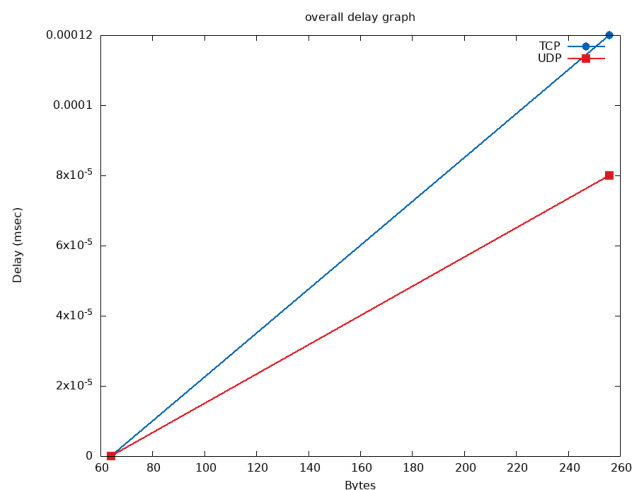
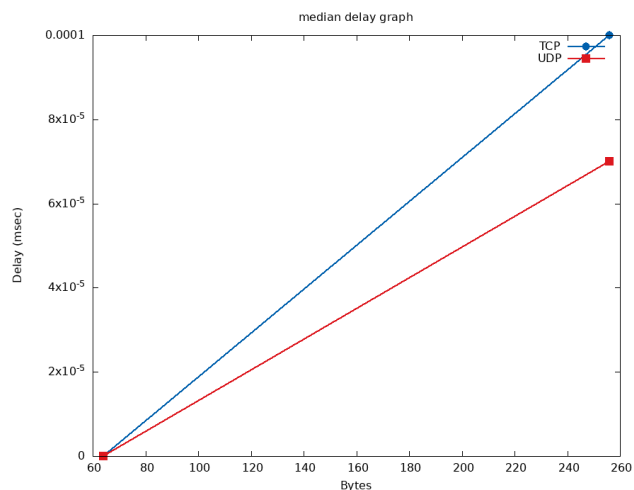
I grafici sono stati creati usando 'gnuplot' tramite uno script bash che calcola il delay.



In questi due grafici sopra l'unica differenza è il tempo di delay che per il TCP aumenta nel grafico a destra. I dati si riferiscono a pacchetti da 32 e 64 bytes in localhost infatti il tempo di delay è molto piccolo.



Questi due grafici sopra si riferiscono alle stesse impostazioni di prima solo che adesso il server a cui stiamo facendo il ping è seti.dibris.unige.it, infatti la differenza che si nota subito è il tempo di delay che è molto più alto. Una cosa che non siamo riusciti a risolvere è il fatto che con 32 bytes in UDP il server ci rifiutava la connessione.



Questi grafici sono stati fatti usando pacchetti da 64 e 256 byte in localhost, si può notare che comunque il TCP ha un delay maggiore in ogni caso rispetto all'UDP.