

# 01QZD

## Laboratorio di Internet e Comunicazioni



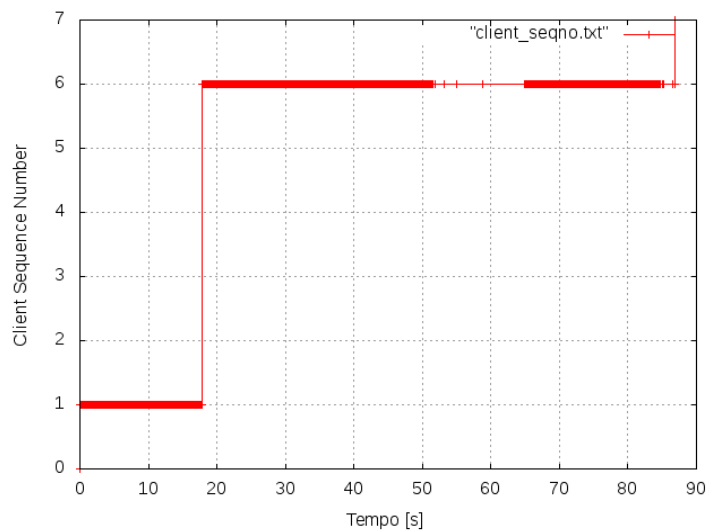
### Internet - Laboratorio 3

#### 4. Analysis of the CHARGEN service

##### 1) Numeri di sequenza del client

Il client inizialmente invia una SYN al server per inizializzare la connessione. Essa ha numero di sequenza relativo pari a 0. Dopo aver ricevuto la SYN ACK da parte del server, il client manda una ACK di risposta con numero di sequenza 1. Da quell'istante il client non trasmette più pacchetti, dunque il numero di sequenza si mantiene costante al valore unitario.

Dopo alcuni secondi, con il comando CTRL+C, si fa in modo che Telnet non effettui stampe su schermo dei pacchetti ricevuti. A questo punto il client manda una PSH ACK di lunghezza 5 byte al server al fine di velocizzare la trasmissione dei pacchetti. Di conseguenza, il numero di sequenza sale a 6. Esso si mantiene costante fino alla chiusura della connessione. In quel momento la FIN ACK mandata dal client causa un aumento del numero di sequenza che si assesta a 7.

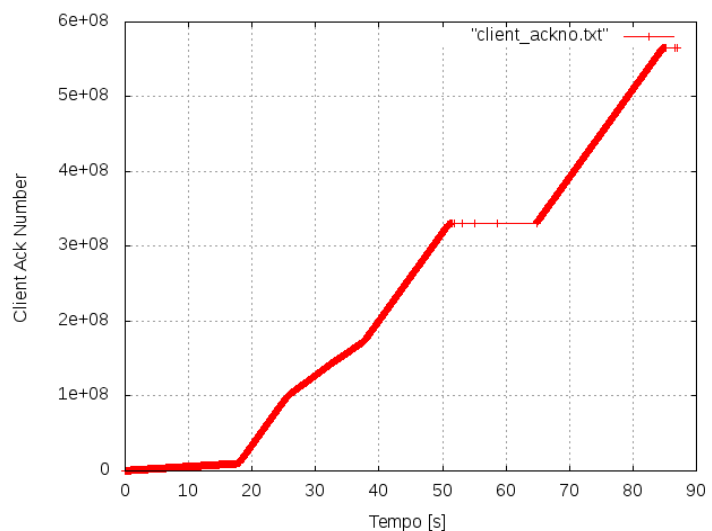


##### 2) Ack Number del client

L'ack number cresce linearmente perché, a parità di condizioni nel tempo, la capacità del client di smaltire i pacchetti si mantiene all'incirca costante.

Dall'istante di apertura della connessione, l'ack number comincia a crescere. Quando si dà il comando per terminare la stampa su schermo (all'istante  $t \approx 18s$ ), la pendenza della retta aumenta considerevolmente poiché i pacchetti vengono trasmessi con frequenza molto più alta.

Successivamente, all'apertura di una seconda connessione, la velocità di trasmissione dei pacchetti diminuisce e

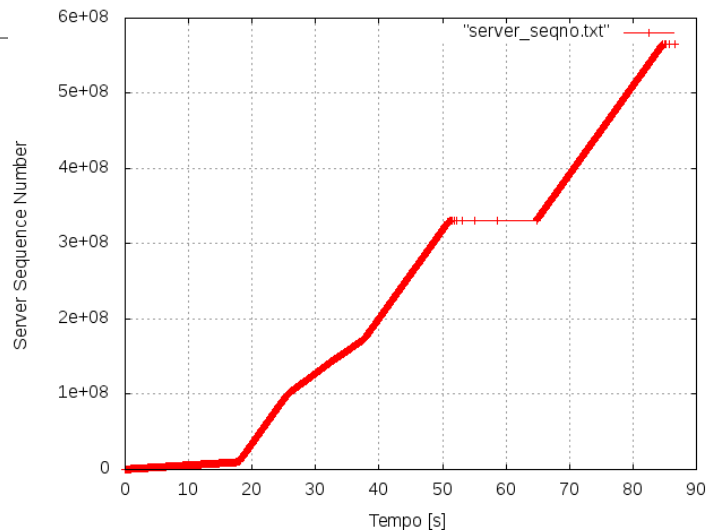


dunque la pendenza diminuisce, per poi ritornare al valore precedente nel momento in cui si chiude la seconda connessione.

Quando si apre la Telnet command mode, la trasmissione di pacchetti viene interrotta, vi sono solo più comunicazioni TCP ZeroWindow a intervalli regolari (con relativa risposta TCP Keep-Alive), e l'ack number si assesta a un valore pressoché costante. Alla chiusura della Telnet command mode, l'ack number torna a crescere e lo fa fino alla chiusura della connessione.

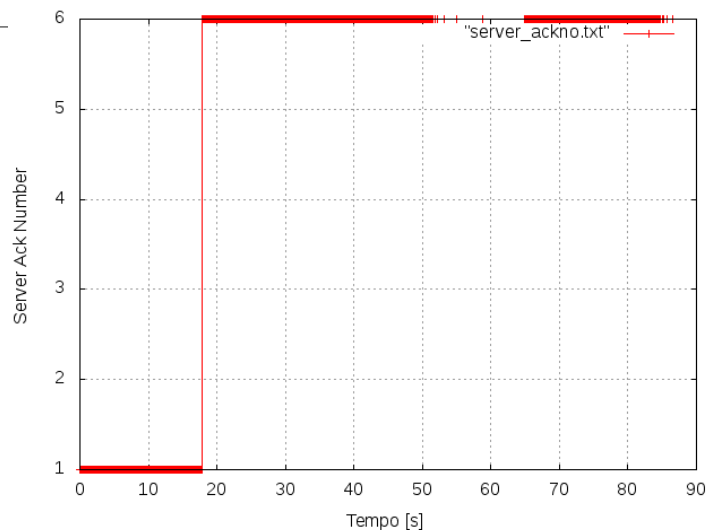
### 3) Server sequence number

L'avanzamento del sequence number del server è del tutto analogo a quello dell'ack number del client



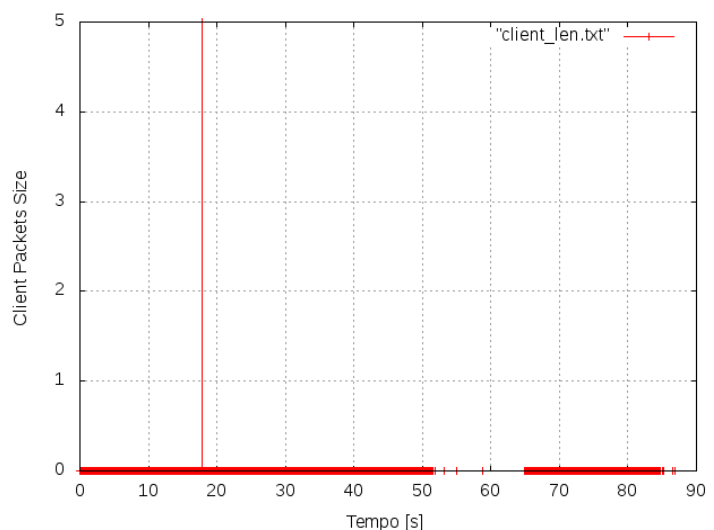
### 4) Server Ack Number

L'avanzamento del server ack number è identico a quello del client sequence number a differenza dell'ultimo pacchetto (quello di chiusura).



### 5) Client Packet Size

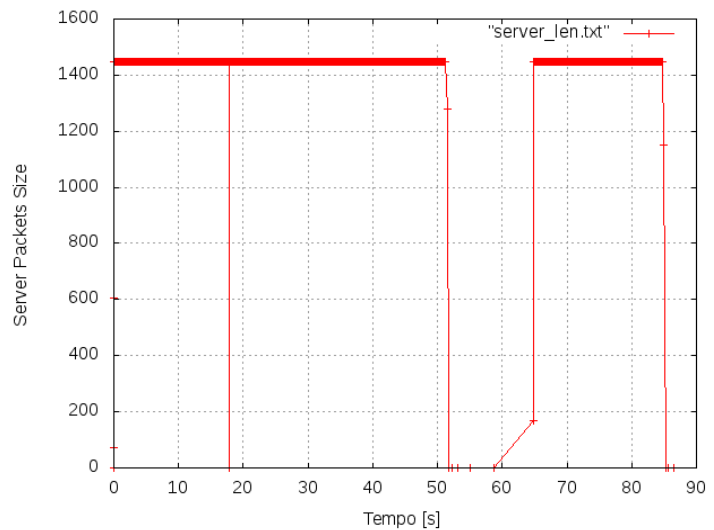
Lungo l'arco di tempo in cui è attiva la connessione, il client trasmette solo ACK, che sono pacchetti a dimensione nulla. Fa eccezione la PSH ACK (di lunghezza 5 byte) trasmessa dal client al server quando si decide di interrompere la stampa dei pacchetti ricevuti dal server.



## 6) Server Packet Size

Dall'istante in cui si apre la connessione il server incomincia a mandare pacchetti di lunghezza pari alla Maximum segment size (1448 bytes).

All'apertura della command mode, il server cessa la trasmissione di pacchetti, per poi riprenderla alla chiusura della stessa, fino al termine della connessione.

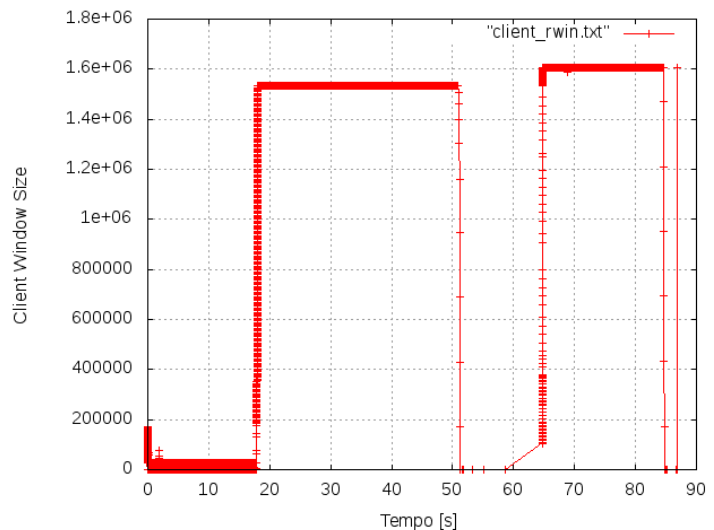


## 7) Finestra di ricezione del client

All'apertura della connessione la finestra del client si apre ad una certa ampiezza. Essa però si restringe immediatamente, in quanto il client, dovendo anche stampare i pacchetti in arrivo, non riesce a smaltire i dati ricevuti dal server.

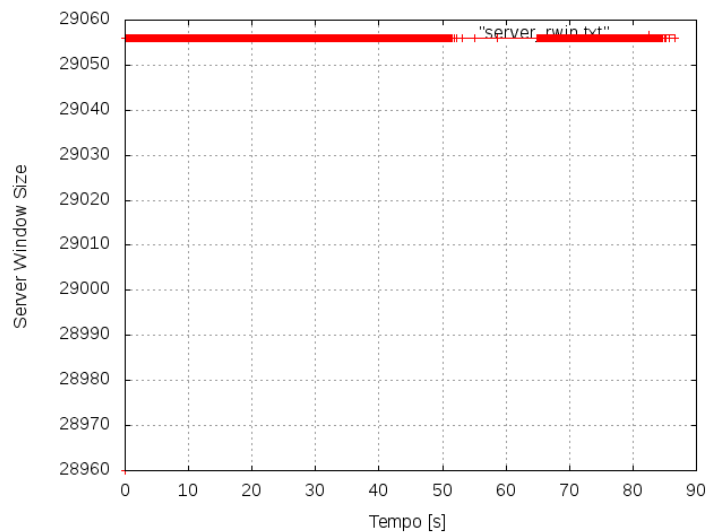
Successivamente la finestra va incontro a una nuova espansione nel momento in cui si ferma la stampa dei pacchetti.

Quando si apre la command mode il client non è più disposto a ricevere dei pacchetti e la finestra si chiude, per poi riaprirsi nel momento in cui si chiude la command mode. Per effettuare la chiusura della connessione, si entra nuovamente nella command mode e la finestra si azzerava. L'ultima apertura della stessa è istantanea, ed è contestuale alla ricezione dell'ACK di chiusura mandata dal server.



## 8) Finestra di ricezione del server

La finestra di ricezione del server rimane costante durante tutta la cattura in quanto non ha bisogno né di ridurla né di aumentarla perchè non ha nessun problema (in capacità di calcolo) di smaltire i pacchetti di ACK.



## Appendice

### processa\_file.sh

```
#!/bin/bash

sed -i "s/^/ /g" client_to_server.txt
sed -i "s/^/ /g" server_to_client.txt

cat client_to_server.txt | grep -v Source | tr -s ' ' | cut -d' ' -f 3 > client.time.txt
cat client_to_server.txt | grep -v Source | cut -d'=' -f 2 | cut -d' ' -f 1 > client.seqno.txt
cat client_to_server.txt | grep -v Source | cut -d'=' -f 3 | cut -d' ' -f 1 > client.ackno.txt.tmp
cat client_to_server.txt | grep -v Source | cut -d'=' -f 4 | cut -d' ' -f 1 > client.rwin.txt.tmp
cat client_to_server.txt | grep -v Source | cut -d'=' -f 5 | cut -d' ' -f 1 > client.len.txt.tmp

cat server_to_client.txt | grep -v Source | tr -s ' ' | cut -d' ' -f 3 > server.time.txt
cat server_to_client.txt | grep -v Source | cut -d'=' -f 2 | cut -d' ' -f 1 > server.seqno.txt
cat server_to_client.txt | grep -v Source | cut -d'=' -f 3 | cut -d' ' -f 1 > server.ackno.txt
cat server_to_client.txt | grep -v Source | cut -d'=' -f 4 | cut -d' ' -f 1 > server.rwin.txt
cat server_to_client.txt | grep -v Source | cut -d'=' -f 5 | cut -d' ' -f 1 > server.len.txt
```

```
# al primo pacchetto manca l'ack e quindi mi prende la lunghezza sbagliata
echo 0 > client.len.txt
tail client.len.txt.tmp -n +2 >> client.len.txt

echo 0 > client.ackno.txt
tail client.ackno.txt.tmp -n +2 >> client.ackno.txt

echo 0 > client.rwin.txt.tmp
tail client.rwin.txt.tmp -n +2 >> client.rwin.txt

rm -rf *.tmp

paste client.time.txt client.seqno.txt > client_seqno.txt
paste client.time.txt client.ackno.txt > client_ackno.txt
paste client.time.txt client.rwin.txt > client_rwin.txt
paste client.time.txt client.len.txt > client_len.txt

paste server.time.txt server.seqno.txt > server_seqno.txt
paste server.time.txt server.ackno.txt > server_ackno.txt
paste server.time.txt server.rwin.txt > server_rwin.txt
paste server.time.txt server.len.txt > server_len.txt

gnuplot client_seqno.plt
gnuplot client_ackno.plt
gnuplot client_rwin.plt
gnuplot client_len.plt

gnuplot server_seqno.plt
gnuplot server_ackno.plt
gnuplot server_rwin.plt
gnuplot server_len.plt
```

---

## client\_seqno.plt

```
set xlabel "Tempo [s]"
set ylabel "Client Sequence Number"
set grid
set terminal png
set output "client_seqno.png"
plot "client_seqno.txt" with linespoint
```

Tutti gli altri file `plt` sono del tutto analoghi, abbiamo solamente cambiato la `ylabel` e i file di I/O