

Homework Multimedia

Lorenzo Mancini 2007951

1 Giugno 2023

Sommario

Report Homework 2: Strumenti per valutazione prestazioni di rete

1 Introduzione

L'obiettivo di questo Homework è quello di effettuare una stima del throughput di una connessione tramite il modello matematico fornito.

2 Istruzioni di esecuzione del codice

Lo sviluppo del progetto è stato effettuato in ambiente MatLab, per cui per poter eseguire il programma bisogna prima importare il progetto nella directory del workspace dell'elaboratore sulla quale si intende operare. Successivamente si procede all'apertura del file e nella sezione "EDITOR" avviare l'esecuzione tramite il tasto "Run".

Affinché il comando `psping` chiamato dalla funzione `system()` di MATLAB possa essere eseguito, bisogna aggiungere alla variabile d'ambiente `PATH` la cartella `PSTools` contenente il file `psping.exe`.

Nota integrativa al codice del programma

Durante lo sviluppo del codice è stato deciso di mantenere il programma il più generale possibile. A questo scopo, il server di destinazione è modificabile prima di avviare l'esecuzione del programma, e successivamente anche il calcolo del parametro `TTL` (dipendente dal server di destinazione) viene eseguito in automatico dall'algoritmo.

Di conseguenza quando il programma viene avviato, i tempi di attesa dei risultati aumentano, in quanto l'elaboratore calcola questi parametri ad ogni esecuzione.

3 Descrizione dei parametri dell'esperimento

Server, numero di istanze `K`, dimensioni dei pacchetti, altre eventuali opzioni del ping.

3.1 Server utilizzato

Il programma creato per questo homework è stato pensato in modo da funzionare indipendentemente dal server prescelto. Infatti è possibile cambiare l'indirizzo del server ad ogni esecuzione senza rendere obsoleto il programma.

Come server di default utilizzato per i test è stato scelto `"atl.speedtest.clouvider.net"`.

3.2 Comprensione del comando `ping`/`psping`

Si verifichi l'esistenza e il formato delle opzioni che consentono di attivare le seguenti funzionalità:

- Definire il numero `K` di pacchetti spediti per ogni sessione ping:
Per entrambi i comandi `ping` e `psping` l'opzione da attivare per questa funzionalità è `'-n'` seguita dal numero di richieste da effettuare. In `psping` si accoda `'s'` al comando per specificare il periodo che deve trascorrere tra una richiesta e l'altra.

```
psping [-n count[s]] destination_server
```

- Specificare la dimensione L_{byte} del payload del pacchetto:
Per entrambi i comandi ping e psping l'opzione da attivare per questa funzionalita' e' '-l' seguita dalla dimensione della richiesta. In psping si accoda 'k' al comando per specificare che la dimensione sia in kilobytes.

```
psping [-l requestsize[k]] destination_server
```

- Definire il Time-To-Live (TTL):
Usufruire di questa funzionalita' utilizzando il psping non e' possibile.
Al contrario utilizzando il comando ping l'opzione da attivare e' '-i' seguita dal TTL.

```
ping [-i TTL] destination_server
```

Per il comando psping l'opzione '-i' e' attribuita ad un'altra funzionalita'.

3.3 Altre eventuali opzioni del comando psping

Per il comando psping sono state utilizzate anche altre funzionalita':

- '-i' opzione per specificare l'intervallo di tempo tra una richiesta e l'altra.

```
psping [-i interval] destination_server
```

- '-q' opzione per non stampare output durante l'esecuzione della richiesta.

```
psping [-q] destination_server
```

- '-4' opzione che specifica l'utilizzo del protocollo IPv4.

```
psping [-4] destination_server
```

4 Stima del numero di link attraversati

Per stimare il numero di link attraversati, indipendentemente dal server scelto, il programma opera in questo modo:

- attraverso un'operazione di ping al server ne conosce l'indirizzo IP, utile per successivi controlli.
- tenta la richiesta al server, partendo con un $TTL=1$ e incrementandolo di una unita' ogni volta che la richiesta risulta fallita, finche' il server non viene raggiunto e la richiesta viene soddisfatta. Memorizzando in questo momento il numero di hop attraversati per raggiungere il server, e quindi il TTL netto.
- determina il TTL netto attraverso il comando tracer

```
tracert -4 destination_server
```

- a questo punto se i due valori del TTL netto non corrispondono l'utente viene avvisato tramite un warning
- infine viene calcolato il TTL andata/ritorno della richiesta, ovvero $n=(2*TTL)-1$.

5 Andamento dell'RTT in funzione della dimensione totale del pacchetto

5.1 RTT minimo

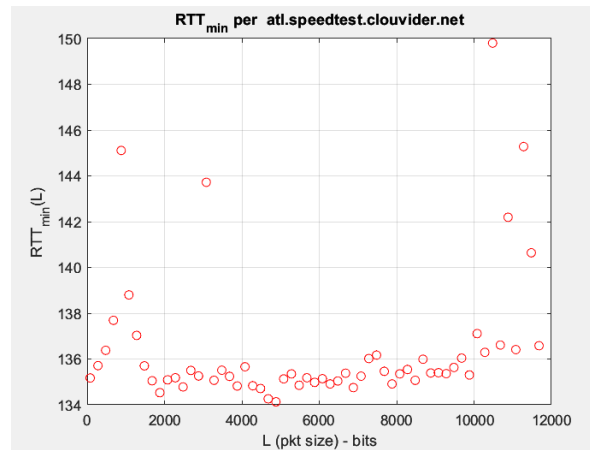


Figura 1: RTT_{min} in funzione di L

5.2 RTT massimo

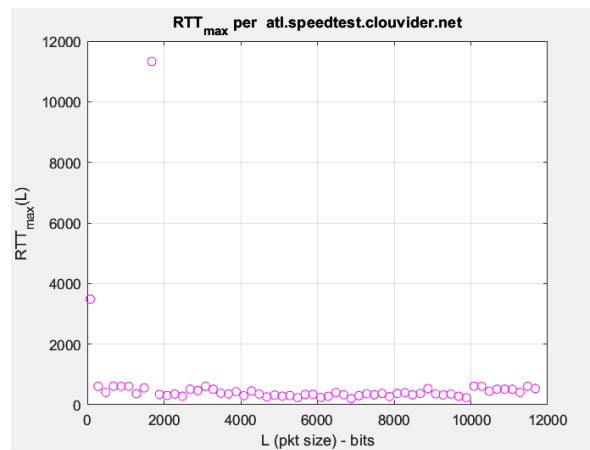


Figura 2: RTT_{max} in funzione di L

5.3 RTT medio

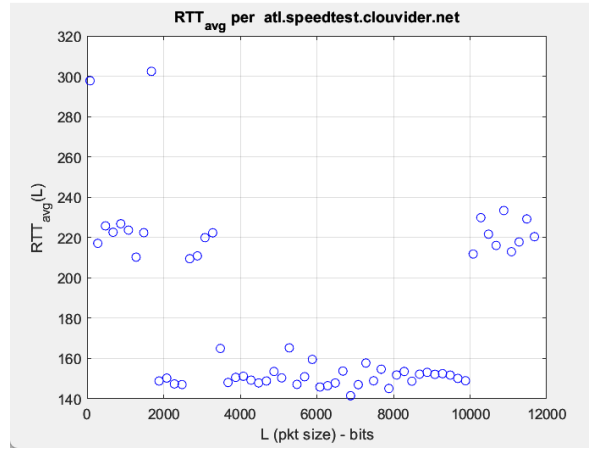


Figura 3: RTT_{avg} in funzione di L

5.4 RTT deviazione standard

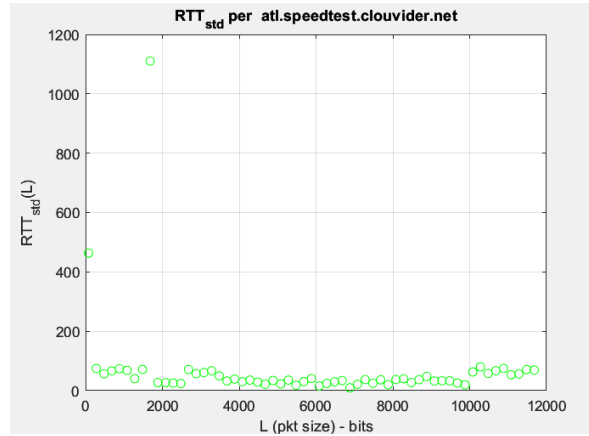


Figura 4: RTT_{std} in funzione di L

6 Stima del throughput

Dalla stima del coefficiente lineare 'a' tramite la funzione polyfit di MATLAB

$$p = \text{polyfit}(pkt_size, RTT_min, 1)$$

$$a = p(1) * 0.001 = 1.13 * 10^7 s/bit$$

che indica la pendenza della retta di fit, sono stati stimati due diversi throughput:

6.1 Throughput con link identici

$$R = n/a = 265 * 10^6 s/bit$$

6.2 Throughput minimo con bottleneck

$$R_{bottleneck} = 2/a = 17 * 10^6 s/bit$$

7 Discussione dei risultati ottenuti

Dai risultati stimati si possono verificare le tre ipotesi:

- Il valore del Round-Trip Time RTT dipende dalla lunghezza L del pacchetto.
- L'andamento minimo del RTT può essere approssimato a una retta.
- Stimando la pendenza di questa retta è possibile ottenere due stime: throughput con link identici e throughput_{bottleneck}.

Notiamo infatti che i valori dei due throughput trovati sono ragionevoli, ovvero un throughput in caso di link identici pari circa a 33 Mbyte/s ed un throughput minimo di bottleneck pari circa a 2.2 MByte/s