

Politecnico di Torino  
III Facoltà di Ingegneria



Corso di Simulazione di Reti e Protocolli

## Simulazione di un Web Server

Fabrizio Agnoli Matr. 151084

Giuseppe Dell'Isola Matr. 151085

# Indice

|   |   |
|---|---|
| 1. Descrizione del Modello di Simulazione ..... | 2 |
| 2. Eventi .....                                 | 2 |
| 3. Misure Prestazionali.....                    | 4 |
| 4. Impatto del Session Rate.....                | 5 |
| 5. Impatto dell'impazienza dei clienti.....     | 6 |
| 5.1 <i>Variazione di BTHRESH</i> .....          | 6 |
| 5.2 <i>Variazione di POLLTIME</i> .....         | 7 |
| 6. Intervalli di Confidenza .....               | 8 |

Lo scopo dell'attività di laboratorio è lo sviluppo di un Web Server che, a fronte di richieste ricevute da un numero potenzialmente infinito di clienti, fornisce file in esso contenuto. Le richieste dei clienti vengono soddisfatte attraverso un unico canale di comunicazione, quindi la sua capacità (1 Mbps) dovrà essere divisa per tutte le richieste che in quel momento il server sta soddisfacendo. I clienti possono effettuare una sola richiesta alla volta (per un massimo di 10 richieste per sessione) e alla fine lasciano il sistema che può essere abbandonato dal cliente anche prima di terminare le sue richieste, in caso di impazienza.

## 1. Descrizione del Modello di Simulazione

Il modello di simulazione è del tipo *Event Driven* ed è stato sviluppato a partire dalle funzioni e sulle strutture dati messe a disposizione durante i laboratori, soprattutto per la gestione degli eventi. Come variabile di stato viene scelto il numero delle richieste attive contemporaneamente sul server ( $n_{rich\_attive}$ ), questo in quanto influenzerà l'utilizzo del link e la gestione (accettazione o rifiuto) di nuove richieste da parte del server.

Non è necessario mantenere una struttura per la memorizzazione delle informazioni dei clienti, mentre per mantenere traccia delle richieste dei clienti è stata usata una struttura Record contenente:

- *arrival*: tempo di arrivo della richiesta nel sistema
- *k*: numero di richieste ancora da soddisfare per la singola sessione
- *band*: banda assegnata alla richiesta
- *fileSize*: grandezza del file interessato al trasferimento
- *fileTx*: quantità di file che è già stata trasmessa
- *t\_time*: tempo residuo per la trasmissione
- *banda\_media*: banda media associata alla richiesta
- *last\_event\_time*: tempo aggiornato ogni volta che c'è qualche cambiamento nel record
- *\*evento*: puntatore all'evento FINE\_RICHIESTA
- *\*eventoControllo*: puntatore all'evento CONTROLLO

Per tenere traccia delle richieste attive non viene usata una coda FIFO dove accodare le richieste che arrivano nel sistema ma si sfruttano i puntatori nel record che puntano agli eventi, e il puntatore presente nella struttura Event che indicherà il record utilizzano la lista lineare bilinkata FES.

## 2. Eventi

Gli eventi del nostro modello di simulazione sono quattro che rappresentano l'arrivo di un cliente nel sistema, l'arrivo di una nuova richiesta, la fine di una richiesta e il controllo sulla banda che viene effettuato ogni intervallo di tempo scandito dal PollTime.

- **ARRIVO CLIENTE**: è l'evento che permette l'inizializzazione del simulatore; modella l'arrivo di nuove sessioni al sistema attraverso una distribuzione di Poisson con tasso  $1/\mu$ . Oltre allo schedulare una nuova sessione con andamento precedentemente citato, prevede il calcolo del numero di richieste  $k$ , che andranno a comporre la sessione. Il valore di  $k$  ha distribuzione di tipo uniforme compresa tra i due estremi (0,10); questo valore verrà passato come parametro alla creazione delle richieste, schedulando l'evento **ARRIVO RICHIESTA** per l'arrivo della prima richiesta della sessione, e decrementato al momento del loro

adempimento corrispondente all'evento **FINE RICHIESTA**. All'arrivo di una nuova sessione viene incrementato un contatore indicante il numero di clienti totali serviti dal nostro web server ( $n\_clienti$ ).

- **ARRIVO RICHIESTA:** in questo evento come prima operazione viene aumentato il contatore  $n\_richieste$  indicante tutte le richieste, sia accettate che rifiutate, ricevute dal sistema. Si va in seguito a vedere se il numero di richieste attive sul server  $n\_rich\_attive$  è inferiore a una data soglia da noi fissata  $MAX\_REQ$ . Il valore simula il numero massimo di richieste gestibile contemporaneamente dal *web server* che stiamo simulando. Nel caso in cui il server ha già raggiunto questo valore limite la richiesta viene rifiutata e termina la sessione associata aggiornando i rispettivi contatori  $n\_richieste\_rifiutate$  e  $n\_sessioni\_incomplete$ ; in caso contrario si aggiorna il numero di richieste attive e viene creato il record per la richiesta andando ad inizializzare i valori visti precedentemente. In particolare, ogni richiesta è relativa ad un singolo file, la cui dimensione viene calcolata attraverso una distribuzione esponenziale negativa con valore medio di 50Kbyte. A seconda della dimensione del file e della banda associata a questa richiesta viene calcolato il valore del tempo "stimato" per la trasmissione che sarà utilizzato per la schedulazione dell'evento di **FINE RICHIESTA**. Essendo che è arrivata una nuova richiesta bisognerà andare a riassegnare la banda a tutte le richieste attive in questo momento, cosa fatta richiamando la procedura **modifica\_banda()** che verrà illustrata in seguito. Infine ci sarà una schedulazione dell'evento **CONTROLLO** ad un tempo  $\pm 5\%$  di un  $POLLTIME$  assegnato, per il controllo sulla banda media assegnata alla richiesta in esame in quel momento.
- **FINE RICHIESTA:** questo evento viene richiamato al momento della fine trasmissione di un file associato alla richiesta. Viene eliminato il record relativo ed eventualmente rimosso l'evento **CONTROLLO**, se già schedulato per un tempo superiore a quello residuo per la fine richiesta. Successivamente si decrementa il numero  $k$  di richieste rimanenti per la sessione, e se è maggiore di zero viene schedulato l'arrivo di una nuova richiesta, altrimenti la sessione termina correttamente e viene incrementato il contatore  $n\_sessioni\_terminate$ . Anche qui verrà richiamata la procedura **modifica\_banda()** in quanto una richiesta ha lasciato il sistema e bisognerà ridistribuire la banda disponibile verso le altre attive.
- **CONTROLLO:** viene fatto un controllo sulla banda media che è stata fornita alla richiesta fino a quel momento. Se questa è maggiore di una soglia, indicata dalla variabile  $BTHRESH$  viene schedulato un nuovo evento di controllo per la stessa richiesta; in caso contrario si effettua un successivo controllo per indicare se il cliente è impaziente o meno. Questo viene fatto attraverso una probabilità calcolata con un uniforme tra 0 e 1, confrontandola con la variabile  $IMP\_PROB$  che rappresenta l'impazienza del cliente con probabilità 10%. Nel caso in cui la probabilità calcolata risulti inferiore a  $IMP\_PROB$  il cliente è considerato impaziente, andando a eliminare l'evento di **FINE RICHIESTA** associato ed incrementando diversi contatori ( $n\_sessioni\_impazienti$ ;  $n\_sessioni\_incomplete$ ;  $n\_richieste\_impazienti$ ). In seguito, anche in questa situazione, bisognerà ridistribuire la banda.

Come visto, la gestione della banda dipende fortemente dal numero di richieste attive sul canale. Occorre quindi prestare particolare attenzione alle variazioni di carico sul web server, rappresentate dal numero di richieste contemporaneamente attive.

Per questo gli eventi **ARRIVO RICHIESTA**, **FINE RICHIESTA** e **CONTROLLO** devono provvedere costantemente al ridimensionamento delle bande associate alla variabile di sistema, *n\_rich\_attive*.

Per fare ciò viene richiamata la funzione **modifica\_banda()**. Questa effettua un nuovo calcolo sulla banda istantanea, in base al numero di richieste attive ed una scansione della FES ricercando tra tutti gli eventi quelli di **FINE RICHIESTA** creando una lista temporanea. Per ogni record puntato da questi eventi contenuti nella lista temporanea *event\_list\_tmp*, si va a verificare la dimensione del file oggetto della richiesta, che è già stato trasmesso. In base a questo valore e al valore della banda dedicata alla richiesta viene calcolato il tempo residuo di trasferimento usato per la schedulazione della nuova **FINE RICHIESTA**.

### 3. Misure Prestazionali

- **Distribuzione del numero medio di richieste concorrenti attive:** Fornisce in un vettore, delle misurazioni sul numero di richieste attive rispetto al tempo di simulazione. Permette di valutare le quantità di tempo in cui il sistema ha un dato numero di richieste attive, da 0 a MAX\_REQ. Le misure sono archiviate in un vettore *Time istogram[MAX\_REQ]*.
- **Numero medio di richieste concorrenti attive:** si tratta di una media temporale. Statistica ottenibile conoscendo la storia del sistema in termini di quantità di tempo per numero richiesta. Il dato è contenuto nel vettore *Time istogram[MAX\_REQ]*; ogni elemento, si tratta di un riferimento temporale espresso in secondi, viene moltiplicato per il numero di richiesta corrispondente. In questo modo si va a comporre una cumulativa espressa in numero richieste per secondi, che permetterà di ottenere la misura prestazionale richiesta con una divisione per il tempo totale di simulazione.
- **Banda media ricevuta dalle richieste:** si tratta di una media ricavabile a partire da una cumulativa delle bande istantanee di ogni richiesta, valida per tutte le richieste. Considerando che la banda per una richiesta non è statica ma inversamente proporzionale al numero di richieste attiva nel sistema occorre sommare ogni contributo, in termini di banda, pesato per il tempo effettivo in cui si ha quella banda. La cumulativa *banda\_media* verrà poi divisa per il numero di richieste accettate dal sistema.
- **Probabilità che una sessione termini prima che tutte le sue richieste siano completate:** media statistica basata su due contatori. *n\_sessioni\_incomplete* viene incrementato ogni volta che una richiesta termina prima di essere completata, caso di impazienza, e anche quando la richiesta viene rifiutata dal web server e *n\_clienti* che rappresenta il numero totale di sessioni instaurate nel sistema.
- **Probabilità che una sessione termini per impazienza dei clienti:** media molto simile alla precedente, con la differenza che il primo contatore, in questo caso *n\_sessioni\_impazienti* tiene conto solamente delle sessioni interrotte per impazienza e non di quelle terminate a causa di un rifiuto del web server.

- **Probabilità che una richiesta termini per impazienza dei clienti:** la differenza con la precedente sta nel fatto che qui non si considerano le sessioni ma le richieste. Un contatore memorizzerà il numero delle richieste rifiutate per la sola impazienza dei clienti ( $n\_richieste\_impazienti$ ) che verrà poi diviso per il numero totale di richieste ricevute dal sistema.
- **Probabilità che una richiesta sia rifiutata:** media che tiene conto delle sole richieste rifiutate a causa di un carico eccessivo sul server, limitato dal valore di MAX\_REQ. Il contatore  $n\_richieste\_rifiutate$  sarà poi sempre pesato sul numero totale di richieste.
- **Server goodput:** variabile che descrive la banda media assegnata alle sole richieste terminate con successo.

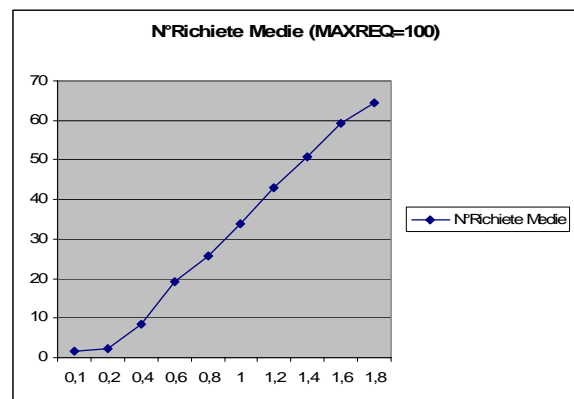
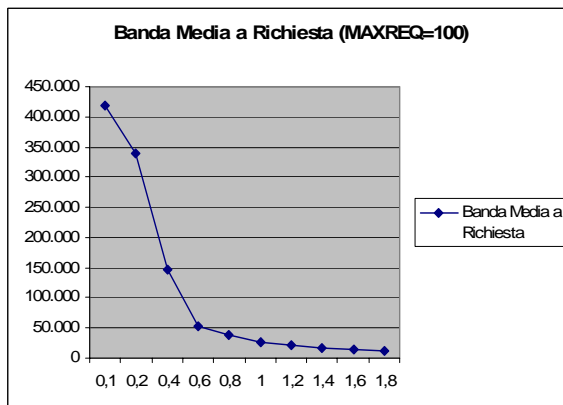
## 4. Impatto del Session Rate

Il session rate è il parametro da cui dipendono gli arrivi nel sistema, ed è quindi fondamentale per la simulazione.

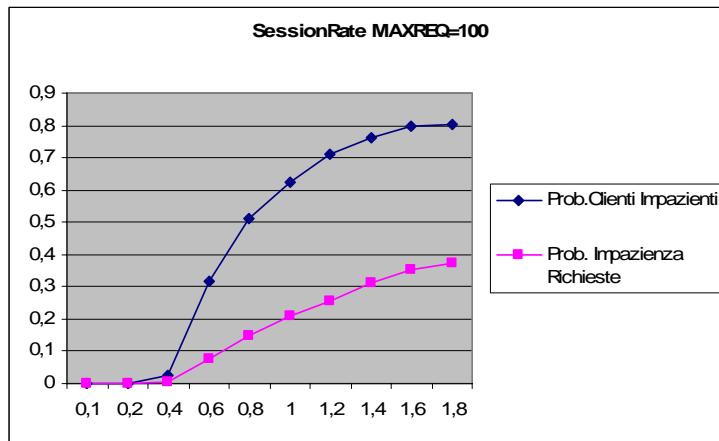
L'analisi viene effettuata variando il session rate; i risultati ottenuti saranno fortemente dipendenti dai valori, scelti da noi a priori, dei parametri che caratterizzano il sistema.

Per questa osservazione è stato scelto un server capace di accettare 100 richieste contemporanee con una **BThresh** pari a 50kbps; **PollTime** di 5 secondi e un tempo di simulazione di 10000.

In seguito verranno riportati i grafici ottenuti con il sistema nelle caratteristiche sopra elencate.



All'aumentare del session rate, come logico aspettarsi, cresce il numero medio di richieste servite in contemporanea dal server con una diminuzione della banda media assegnata. Si noti come la banda media raggiunga valori prossimi alla BThresh rapidamente intorno a valore del session rate di 0.6, corrispondente a circa 20 richieste, per poi scendere ulteriormente al di sotto della BThresh fino a valori di session rate di 1.8 corrispondenti a circa 65 richieste medie.



Da questo grafico si può notare come le probabilità di scarto di sessioni e richieste crescano con l'aumentare del session rate. Logicamente la probabilità di clienti impazienti cresce molto più rapidamente rispetto a quella associata alle richieste in quanto la sessione è formata da più richieste, fino a un massimo di 10.

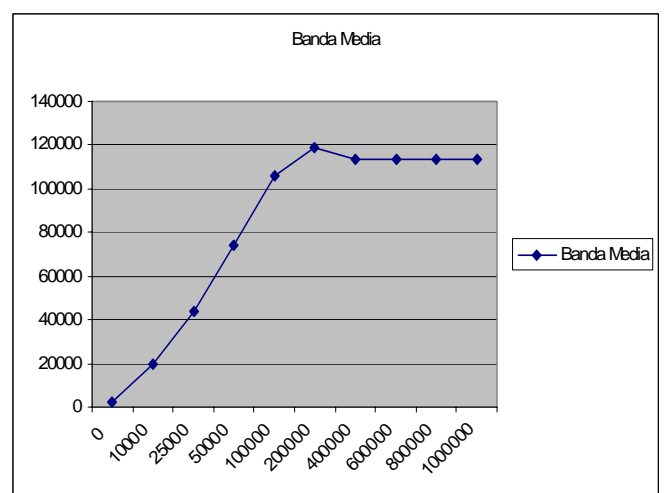
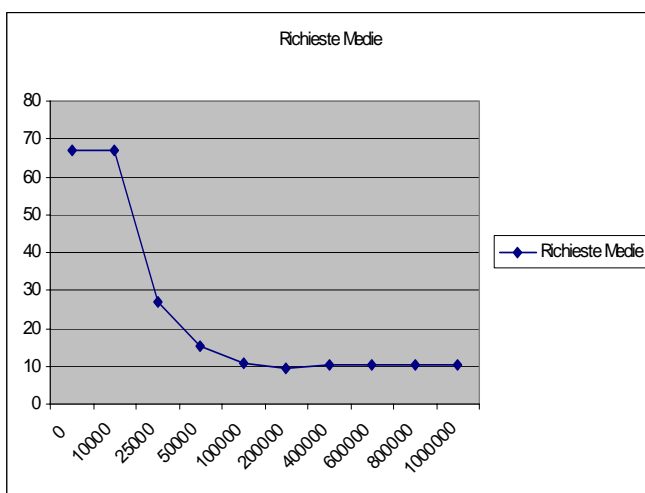
## 5. Impatto dell'impazienza dei clienti

### 5.1 Variazione di BTHRESH

La **BThresh** è il parametro che rappresenta una soglia, sotto la quale i clienti hanno probabilità **IMP\_PROB** di essere impazienti e quindi di interrompere immediatamente la richiesta e di conseguenza l'intera sessione.

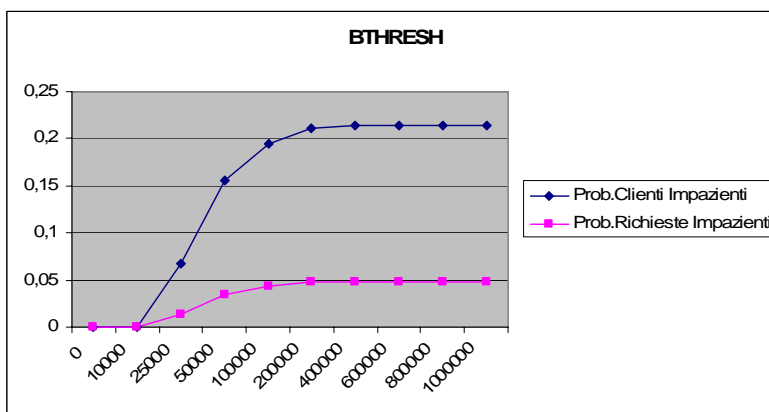
Per questo scenario si sono scelti valori di simulazione corrispondenti ad un **session rate** di 0.5, **PollTime** di 5s, **MAX\_REQ** di 100, **IMP\_PROB** di 0.1 e un tempo di simulazione di 10000.

Per valori di BThresh crescenti abbiamo ottenuto i seguenti risultati



Si può notare come al crescere della **BThresh** si riscontra una diminuzione del numero medio di richieste servite contemporaneamente dal web server con un conseguente aumento della banda media a loro assegnata. Dopo una diminuzione molto rapida del numero di richieste si giunge ad un asintoto orizzontale intorno alle 10 richieste; valore raggiunto per valori di BThresh di 400Kbps fino a 1Mbps (capacità massima del link).

Dal secondo grafico emerge come la banda media cresca rapidamente fino ai 400Kbps di BThresh per assestarsi intorno ai 110Kbps.

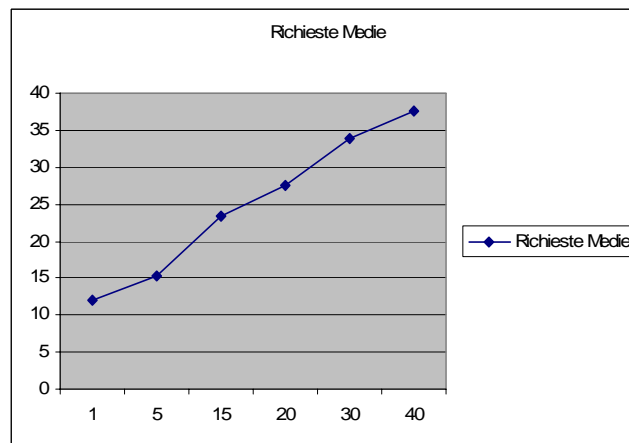
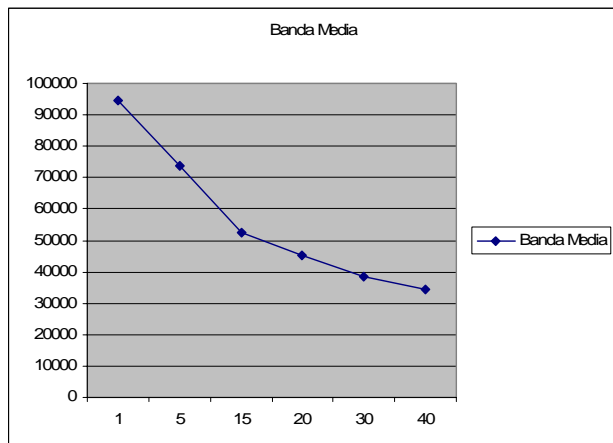


Col crescere della BThresh aumenta la probabilità di impazienza dei clienti, molto rapidamente per valori più bassi di banda, per poi stabilirsi intorno al 21%. Stesso comportamento si riscontra per le richieste impazienti. Anche in questo caso si noti come la crescita della probabilità delle richieste impazienti sia più bassa rispetto alla probabilità dei clienti impazienti.

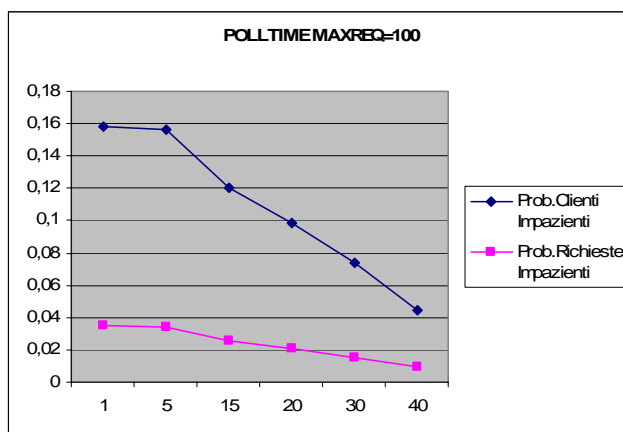
## 5.2 Variazione di POLLTIME

Il secondo studio sull'impazienza dei clienti viene fatta modificando il valore del **PollTime** che indica il tempo con cui viene effettuato il controllo sulla banda media associata alla richiesta, rispetto alla **BThresh**. Per questa analisi si è scelto un valore di BThresh di 50kbps e un **IMP\_PROB** di 0.1; oltre a un **session rate** di 0.5, un **MAX\_REQ** di 100 e un tempo di simulazione pari a 10000.

Si è scelto di far variare il valore di PollTime da 1s fino a un massimo di 40, ottenendo i seguenti risultati.



Al crescere del PollTime il numero di richieste medie servite in contemporanea dal sistema cresce, in quanto viene effettuato un controllo sulla banda sempre più lontano nel tempo rispetto al tempo di arrivo della richiesta. La banda media, invece, logicamente diminuisce al crescere del PollTime.



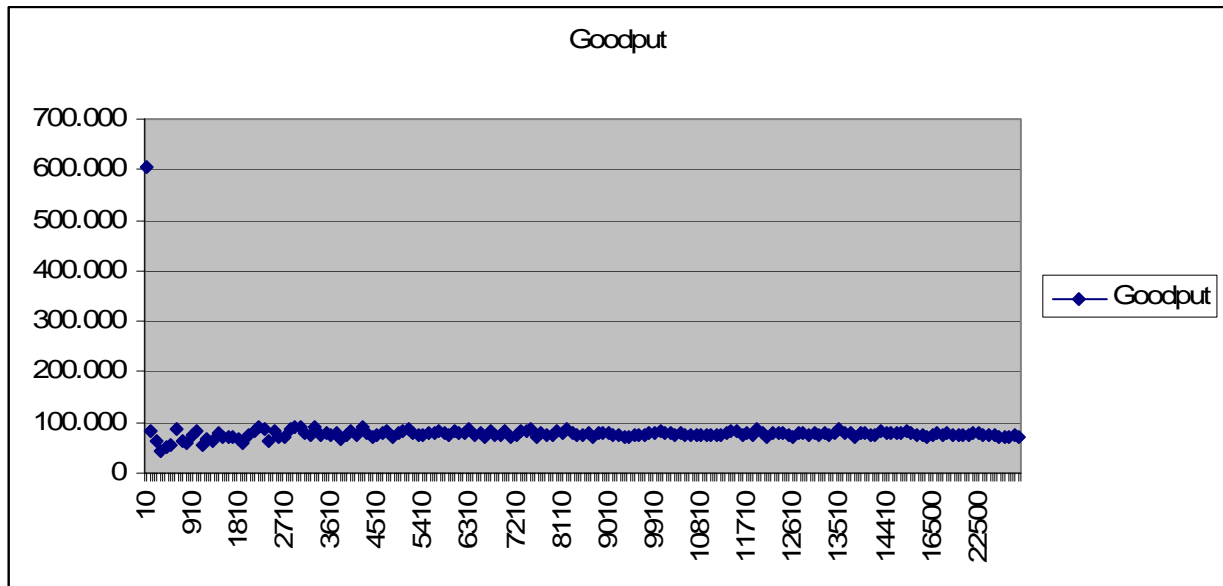
Da queste prove emerge, come lecito aspettarsi, che all'aumentare del PollTime diminuisce la probabilità di scarto per impazienza, perché la condizione che la banda media attuale sia superiore alla BThresh viene controllata in intervalli temporali più ampi, quindi la richiesta viene potenzialmente terminata più tardi con le conseguenze visibili in termini di banda media, che come visto in precedenza, diminuisce.

## **6. Intervalli di Confidenza**

Per il calcolo degli intervalli di confidenza, tra le varie tecniche possibili, abbiamo scelto quella dei *batch means*. La tecnica si basa sui dati raccolti tramite un pool di 10 o più simulazioni.

Il parametro osservato in questo studio è il Server Goodput, che rappresenta la banda media associata alle richieste terminate con successo, valore generalmente leggermente più alto rispetto alla banda media associata a tutte le richieste.

Questo valore è calcolato variando il tempo di simulazione da 0 a 40000 con un session rate di 0.5.



Dal grafico si può notare come il sistema abbia un transitorio sulla fase iniziale che si esaurisce intorno ai 5000s simulati.

Per il calcolo degli intervalli di confidenza questo transitorio non viene considerato,svolgendo 10 simulazioni negli intervalli indicati in tabella.

| DA [s] | A [s] | Goodput Medio [Kbps] |
|--------|-------|----------------------|
| 5000   | 7500  | 76,765               |
| 7500   | 10000 | 77,342               |
| 10000  | 12500 | 77,803               |
| 12500  | 15000 | 76,828               |
| 15000  | 17500 | 78,286               |
| 17500  | 20000 | 77,852               |
| 20000  | 22500 | 77,976               |
| 22500  | 25000 | 77,219               |
| 25000  | 27500 | 77,688               |
| 27500  | 30000 | 78,238               |

Ottenuti così gli intervalli si passa al calcolo della media aritmetica e della varianza delle medie per intervallo analizzato,utilizzando le formula seguenti

$$m = \frac{1}{n} \sum_{i=1}^n x_i \qquad s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - m)^2}$$

Otteniamo dei valori di media  $m=77,693[kbps]$  e varianza  $s^2=0,302$ .

Si procede al calcolo degli intervalli di confidenza,siccome il numero di simulazioni è inferiore a 30 ci avvaliamo dalla tabella per le distribuzioni di *Student* con  $n=9$ .

Gli intervalli verranno calcolati con la seguente formula

$$I = \left[ m - t_{9,1-\frac{\alpha}{2}} \cdot \frac{s}{\sqrt{10}}; \quad m + t_{9,1-\frac{\alpha}{2}} \cdot \frac{s}{\sqrt{10}} \right]$$

| <b>Livello di Confidenza</b> | <b>Estremo Inferiore [Kbps]</b> | <b>Estremo Superiore [Kbps]</b> |
|------------------------------|---------------------------------|---------------------------------|
| 99%                          | 77,128                          | 78,257                          |
| 98%                          | 77,203                          | 78,182                          |
| 95%                          | 77,300                          | 78,085                          |
| 90%                          | 77,375                          | 78,010                          |
| 80%                          | 77,453                          | 77,932                          |