ANALISI DELLE CODE PER L'INGRESSO AD UN CONCERTO

Relazione sul progetto per il corso di Performance Modeling of Computer Systems and Networks

Danilo D'Amico

Matricola 0320389

Università di Roma Tor Vergata

SOMMARIO

Presentazione del caso di studio 1
Obiettivi 2
Modello Concettuale 3
    Stati 4
    Eventi 4
Modello delle Specifiche 6
    Distribuzioni 6
Modello Computazionale 9
    Eventi 9
    Descrizione del codice 10
Verifica e Validazione 13
    Verifica 13
    Validazione 14
Simulazione a Orizzonte Finito 15
Simulazione a Orizzonte Infinito 19

C a p i t o l o 1

PRESENTAZIONE DEL CASO DI STUDIO

Questo progetto punta a modellare la gestione degli ingressi in una sala per concerti tramite una rete di code. Sebbene la struttura e le code immaginate siano ideali, cercano di ricalcare nei dati disponibili il Palazzo dello Sport di Roma.

Si immagina di modellare il caso di un concerto *sold out* in cui tutti i posti vadano occupati prima dell'inizio del concerto. I posti in platea sono stati considerati come posti VIP che consentono l'ingresso privilegiato nella struttura.

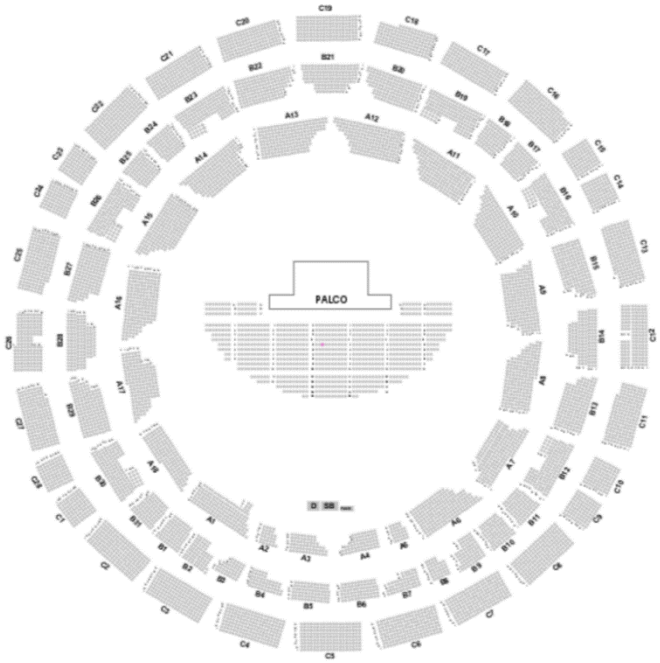


Figura 1 Disposizione dei posti a sedere nel Palazzo dello Sport (Roma) per un concerto di Claudio Baglioni

Capitolo 2

OBIETTIVI

L'obiettivo dello studio è creare una simulazione *next-event* che modelli il passaggio degli utenti attraverso le varie code del sistema in modo realistico minimizzando il numero dei serventi per smaltire gli ingressi garantendo i seguenti livelli di servizio per l'attesa media in ogni coda:

1. L'attesa media per l'acquisto di un biglietto deve essere inferiore ai 3 minuti

2. L'attesa media per una perquisizione deve essere inferiore ai 10 minuti

3. L'attesa media per la validazione di un ticket per un posto in platea deve essere inferiore al minuto

4. L'attesa media per la validazione di un ticket per un posto non in platea deve essere inferiore ai 10 minuti

5. L'attesa media per la coda per il Backstage deve essere inferiore ai 5 minuti. In questo caso la variabile da gestire non è il numero dei serventi (che è solo l'artista) ma il tempo che si può concedere ad ogni spettatore

C a p i t o l o 3

MODELLO CONCETTUALE

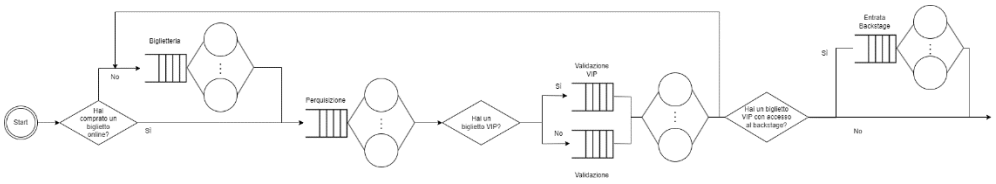


Figura 2 Diagramma del sistema

Il diagramma in figura rappresenta il modello concettuale del sistema. A partire da *Start* a sinistra, il primo diamante divide gli arrivi tra il 75% che ha già acquistato il biglietto online, mandato direttamente alla fase di perquisizione, ed il 25% che va in biglietteria. A seconda del biglietto acquistato si viene smistati in una di due code per la validazione del biglietto: una coda prioritaria VIP ed una coda normale. In caso di non validità del biglietto (nominativo sbagliato, documento non valido o biglietto non valido) si viene rimandati in biglietteria. Una volta passata la fase di validazione, agli utenti che hanno acquistato un biglietto VIP con *meet-and-greet* viene concessa l'entrata nel backstage.

Ciascun server modella un impiegato della struttura addetto, rispettivamente, alla vendita dei biglietti, alla perquisizione, alla validazione e all'ingresso nel backstage.

Stati

Ciascun server può essere in uno stato IDLE o BUSY. Ciascuna coda può essere in uno stato EMPTY o NON-EMPTY.

Lo stato iniziale e finale di ogni server sarà IDLE e quello di ogni coda EMPTY.

Eventi

Biglietteria:

- Arrivo di un utente
- Partenza di un utente servito

Perquisizione:

- Arrivo di un utente
- Partenza di un utente servito

Validazione:

- Arrivo di un utente
- Arrivo di un utente VIP
- Partenza di un utente servito con successo
- Partenza di un utente il cui biglietto non è stato validato con successo (feedback verso la Biglietteria)

Backstage:

- Arrivo di un utente

- Partenza di un utente servito

Il sistema viene immaginato con un comportamento *work-conserving*: questo significa che in ogni istante della simulazione nessun servente può essere IDLE se una delle code che serve è NON-EMPTY.

Lo *scheduling* dei membri di una coda è effettuato secondo la politica FIFO e suppone che le code abbiano capacità infinita.

Il servizio dei *job* è *non-preemptive*.

Capitolo 4

MODELLO DELLE SPECIFICHE

Seguendo quanto scritto nel modello Concettuale, si può modellare ogni servente come in uno stato 0 o 1 ad ogni coda viene associato un numero pari ai job arrivati nel sistema e che devono ancora essere consegnati ad un servente.

Distribuzioni

Gli arrivi sono indipendenti tra loro ed il sistema non ha memoria, cioè le informazioni passate non hanno impatto sugli arrivi passati e futuri. Gli arrivi costituiscono un processo di Poisson ed è stata pertanto scelta la distribuzione esponenziale per modellare i tassi di interarrivo.

Per gli stessi motivi anche i tempi di servizio seguono la distribuzione esponenziale.

Il feedback verso la biglietteria, la probabilità di aver comprato un biglietto online e la probabilità di entrare nel backstage sono invece scelti mediante una probabilità costante.

$$p_{feedback} = probabilità\ avere\ un\ biglietto\ non\ valido = 0.06$$

La capienza del Palazzo dello Sport è di 11500 posti, di cui 814 in platea, che vengono considerati VIP in quanto più vicini al palco. Ne segue che la probabilità che un biglietto sia per la platea è del 7,07826%.

$$p_{vip} = probabilità\ di\ avere\ un\ biglietto\ per\ la\ platea = 0.0707826$$

I posti per il *meet&greet* sono stati stimati in 50, quindi la probabilità che si sia acquistato un biglietto in platea con questa opzione è del 6,14251%.

$$p_{backstage} = probabilità\ avere\ un\ biglietto\ per\ il\ backstage$$
$$= 0.0614251$$

Infine, per quanto riguarda la percentuale dei biglietti acquistati online, dal 1° luglio 2019 è in vigore l'obbligo del biglietto nominativo per l'accesso agli eventi con capienza superiore a 5000 spettatori, il che ha aumentato drammaticamente la percentuale di biglietti venduti online e perciò si è deciso per una divisione tra acquisti online e in loco del 75%/25%.

$$p_{online} = probabilità\ di\ aver\ comprato\ il\ biglietto\ online = 0.75$$

Seguono i modelli adottati per le code del sistema:

- Biglietteria: M|M|7, E(S) = 20s
- Perquisizione: M|M|16, E(S) = 15s
- Validazione: M|M|10, E(S) = 10s
- Backstage: M|M|1, E(S) = 28s

Per quanto riguarda gli arrivi, considerando che la struttura apre tre ore prima dell'inizio dell'evento, il tasso di arrivo sarà:

$$\lambda = \frac{11500\ \text{persone}}{180\ \text{minuti}} = 1.06481\ persone/s$$

Equazioni di traffico:

$$\lambda_{biglietteria} = \lambda * (1 - p_{online}) + p_{feedback} * \lambda_{validazione}$$

$$\lambda_{perquisizione} = \lambda_{biglietteria} + \lambda * p_{online}$$

$$\lambda_{validazione} = \lambda_{perquisizione}$$

$$\lambda_{validazione\ vip} = \lambda_{perquisizione} * p_{vip}$$

$$\lambda_{validazione\ non\ vip} = \lambda_{perquisizione} * (1 - p_{vip})$$

$$\lambda_{backstage} = \lambda_{validazione} * p_{backstage} * (1 - p_{feedback})$$

Viene riportato lo pseudocodice per la gestione della validazione:

```
while(true){
    scheduleResult;
    if (!CodaVIP.isEmpty()){
        scheduleResult = schedule first vip in queue
    } else if(!Coda.isEmpty()){
        scheduleResult = schedule first client in queue
    } else{
        goTo next iteration;
    }

    if(scheduleResult is successful){
        move job forward
    } else{
        send job back to Biglietteria
    }
}
```

C a p i t o l o 5

MODELLO COMPUTAZIONALE

Per costruire la simulazione si è fatto uso del materiale messo a disposizione dal Professor Lawrence M. Leemis sul suo sito, autore del libro *Discrete-Event Simulation – A First Course* usato durante il corso.

Il linguaggio scelto per lo sviluppo è stato Java, scelto sia per la disponiblità in tale linguaggio di *snippet* di codice sul sito del Professor Leemis che per la familiarità sviluppata durante altri progetti con il linguaggio.

Eventi

Il modello mantiene un array contenente tutti gli eventi generati ma non ancora avvenuti, diretta implementazione di quanto teorizzato nel modello concettuale. L'array *event* ha una entry per ogni arrivo, partenza e server, così da poter tenere traccia per ognuno di essi dell'orario della prossima partenza.

Segue una tabella con la composizione dell'array nel dettaglio:

Dimensione area	Evento
[1]	Arrivo in Biglietteria da evento generato
[1]	Arrivo in Biglietteria da feedback
[7]	Server Biglietteria
[1]	Arrivo in Perquisizione
[16]	Server Perquisizione

[1]	Arrivo di utente VIP in Validazione
[1]	Arrivo di utente non VIP in Validazione
[10]	Server Validazione
[1]	Arrivo nel Backstage
[1]	Server Backstage

Un evento è modellato dalla classe:

```
class MsqEvent{                 /* the next-event list      */
  double t;                     /*   next event time        */
  int    x;                     /*   event status, 0 or 1   */
  int isPriority;               /* used for priority queues */
}
```

Descrizione del codice

Il modello computazionale è contenuto nel file *ModelloComputazionale.java* del progetto disponibile su GitHub all'indirizzo: https://github.com/DaniloDamico/ProgettoPMCSN

Gli attributi all'interno della classe rappresentano se la cella contiene un evento da considerare o meno (x), in caso affermativo in quale istante si verificherà (t) ed un parametro *isPriority* usato per distinguere gli arrivi prioritari da quelli non prioritari nelle coda con priorità della Validazione.

L'implementazione delle code è effettuata tramite una simulazione *next-event*, lo scheduler sfrutta cioè la classe *MsqT* contenente il timestamp dell'attuale evento e del prossimo come supporto per far avanzare il tempo dall'evento corrente al successivo.

Il file contiene anche una classe e del codice per la raccolta e l'analisi delle statistiche sull'esecuzione della simulazione.

La simulazione termina quando il tempo dell'evento corrente è maggiore uguale a quello di termine (o sono stati raggiunti gli 11500 ingressi) e non ci sono più job sulle code. Superato l'istante di fine della simulazione non vengono più generati nuovi arrivi all'ingresso del sistema:

```
while ((event[0].x != 0) || (numberTicket + numberSearch +
numberValidation + numberVIPValidation + numberBackstage !=
0)) { ...}
```

Il server scelto per servire il prossimo job dalla coda è quello che è stato IDLE più a lungo. Viene riportata la logica usata nella logica relativa alla biglietteria, ma la logica relativa ai nodi successivi è analoga:

```
  int findOneTicket(MsqEvent [] event) {
/* ------------------------------------------------------
 * return the index of the available ticket server idle
longest
 * ------------------------------------------------------
 */
    int s;
    int i = Events.ARRIVAL_TICKET;

    while (event[i].x == 1)        /* find the index of the
first available */
      i++;                         /* (idle) server
*/
    s = i;
    while (i < Events.SERVERS_TICKET + Events.ARRIVAL_TICKET)
{         /* now, check the others to find which   */
      i++;                         /* has been idle longest
*/
      if ((event[i].x == 0) && (event[i].t < event[s].t))
        s = i;
    }
    return (s);
  }
```

I numeri pseudorandomici sono generati tramite un generatore di Lehmer contenuto nel file *Rngs.java*. È stato utilizzato un unico generatore con più stream, uno per ogni processo stocastico per garantire l'*uncoupling* tra ognuna delle generazioni. Nell'assegnazione di uno stream ad un processo questi sono stati distribuiti in modo da massimizzare la distanza tra i 256 stream offerti dalla libreria.

Capitolo 6

VERIFICA E VALIDAZIONE

Verifica

L'obiettivo della fase di verifica è controllare che la simulazione si svolga secondo i modelli descritti finora.

I dati riportati nella seguente tabella, ad esempio, sono facilmente verificabili: il tempo di servizio medio è dato e rispettato, il tempo di interarrivo è l'inverso dei tassi di arrivo calcolati dalle equazioni di traffico scritte nel modello concettuale e l'utilizzo è il quoziente dei due valori e del numero di serventi

Risultati simulativi	Tempo di servizio medio: $E(S)$	Tempo di interarrivo	Utilizzo: ρ
Biglietteria	19,62349 +/- 0,34662	3,15568 +/- 0,04735	0,83874 +/- 0,01536
Perquisizione	14,93772 +/- 0,12997	0,93523 +/- 0,00812	0,93587 +/- 0,00533
Validazione	9,98857 +/- 0,08844	1,07771 +/- 0,01055 (non vip) 13,72784 +/- 0,44116 (vip)	0,88556 +/- 0,00393
Backstage	27,31522 +/- 3,75688	297,91480 +/- 49,25011	0,17987 +/- 0,03173

Sono stati fatti anche i controlli di consistenza $E(T_s) = E(T_q) + E(S_i)$ ed $E(N_s) = E(N_q) + m * \rho$ che hanno dato esiti positivi

Validazione

Purtroppo non è stato possibile fare un raffronto diretto con i dati reali della struttura. Per validare il sistema ci si è limitati a verificare che il comportamento del sistema sia consistente con il comportamento che ci si aspetterebbe da una struttura reale.

Si è constatato che i tempi di servizio crescono al crescere del tasso di arrivo o al diminuire dei serventi in un nodo, che diminuiscono all'avvenire dell'opposto e che le metriche considerate peggiorano o migliorano di conseguenza.

Capitolo 7

SIMULAZIONE A ORIZZONTE FINITO

La simulazione a orizzonte finito è stata svolta usando 1000 replicazioni con la configurazione standard del sistema esposta nei modelli precedenti. I risultati di ogni simulazione sono stati salvati su un array apposito e poi salvati in file *.dat* analizzati da *Estimate.java*.

Data l'applicazione iterativa dell'algoritmo indicato nel libro *Discrete-Event Simulation - A first course*, i parametri usati nelle simulazioni hanno influenzato quelli indicati nei modelli: i parametri dei modelli cioè sono frutto anche dell'applicazione delle simulazioni a orizzonte finito per rispettare i livelli di servizio minimo indicati negli obiettivi.

Seguono i risultati, calcolati con un intervallo di confidenza del 95%:

Biglietteria

Metrica	Risultato Simulativo
Tempo di risposta: $E(T_s)$	50,19 +/- 0,98
Tempo di attesa: $E(T_q)$	30,20 +/- 0,97
Utilizzo: ρ	0,72 +/- 0,00
Popolazione media: $E(N_s)$	14,77 +/- 0,29
Tempo di interarrivo	3,40 +/- 0,00
Tempo di servizio medio: $E(S)$	20,00 +/- 0,02

Popolazione media nella coda: $E(N_q)$	8,90 +/- 0,29

Perquisizione

Metrica	Risultato Simulativo
Tempo di risposta: $E(T_s)$	312,27 +/- 4,72
Tempo di attesa: $E(T_q)$	297,27 +/- 4,72
Utilizzo: ρ	0,93 +/- 0,00
Popolazione media: $E(N_s)$	310,82 +/- 4,71
Tempo di interarrivo	1,00 +/- 0,00
Tempo di servizio medio: $E(S)$	15,00 +/- 0,01
Popolazione media nella coda: $E(N_q)$	295,90 +/- 4,70

Validazione per utenti non in platea

Metrica	Risultato Simulativo
Tempo di risposta: $E(T_s)$	415,38 +/- 5,53
Tempo di attesa: $E(T_q)$	406,38 +/- 5,53
Popolazione media: $E(N_s)$	383,50 +/- 4,97
Tempo di interarrivo	1,08 +/- 0,00
Popolazione media nella coda: $E(N_q)$	375,18 +/- 4,97

Validazione biglietti per la platea

Metrica	Risultato Simulativo
Tempo di risposta: $E(T_s)$	11,08 +/- 0,02
Tempo di attesa: $E(T_q)$	2,06 +/- 0,01
Popolazione media: $E(N_s)$	0,78 +/- 0,00
Tempo di interarrivo	13,98 +/- 0,03
Popolazione media nella coda: $E(N_q)$	0,15 +/- 0,00

Valori globali per i server di validazione

Metrica	Risultato Simulativo
Utilizzo: ρ	0,90 +/- 0,00
Tempo di servizio medio: $E(\mathrm{S})$	10,00 +/- 0,01

Backstage

Metrica	Risultato Simulativo
Tempo di risposta: $E(T_s)$	31,47 +/- 0,36
Tempo di attesa: $E(T_q)$	3,71 +/- 0,17
Utilizzo: ρ	0,13 +/- 0,00
Popolazione media: $E(N_s)$	0,14 +/- 0,00
Tempo di interarrivo	250,93 +/- 2,24

Tempo di servizio medio: $E(\mathrm{S})$	27,76 +/- 0,25
Popolazione media nella coda: $E(N_q)$	0,02 +/- 0,00

Dai valori si può vedere che il nodo con valori di attesa peggiori è quello per la validazione, in modo specifico per gli utenti senza un posto in platea in quanto, insieme al nodo Perquisizione, è l'unico nodo attraverso il quale tutti i job devono passare almeno una volta.

Notiamo inoltre che tutti i valori di utilizzo ρ sono inferiori a 1, il che significa che il sistema opera in equilibrio stocastico.

Capitolo 8

SIMULAZIONE A ORIZZONTE INFINITO

La simulazione a orizzonte infinito è stata svolta con batch means con k = 64 e b = 256. Il sistema non riflette più il sistema reale, il quale aveva una quantità massima di job in ingresso limitata a 11500 (il numero dei posti in sala). Il seed è 456.

Il numero di serventi rispecchia quello individuato come minimo per garantire i livelli di servizio durante la simulazione a orizzonte finito.

Seguono i dati raccolti ed elaborati mediante la stessa classe *Estimate* utilizzata per la simulazione a Orizzonte finito

Biglietteria

Metrica	Risultato Simulativo
Tempo di risposta: $E(T_s)$	41,92215 +/- 6,17192
Tempo di attesa: $E(T_q)$	22,28798 +/- 5,98115
Utilizzo: ρ	0,83874 +/- 0,01536
Popolazione media: $E(N_s)$	12,60067 +/- 1,91160
Tempo di interarrivo	3,15568 +/- 0,04735

Tempo di servizio medio: $E(\mathrm{S})$	19,62349 +/- 0,34662
Popolazione media nella coda: $E(N_q)$	6,72952 +/- 1,84351

Perquisizione

Metrica	Risultato Simulativo
Tempo di risposta: $E(T_s)$	1093,19468 +/- 157,88022
Tempo di attesa: $E(T_q)$	1078,25270 +/- 157,86979
Utilizzo: ρ	0,93587 +/- 0,00533
Popolazione media: $E(N_s)$	1095,72122 +/- 155,70403
Tempo di interarrivo	0,93523 +/- 0,00812
Tempo di servizio medio: $E(\mathrm{S})$	14,93772 +/- 0,12997
Popolazione media nella coda: $E(N_q)$	1080,74729 +/- 155,69745

Validazione per utenti non in platea

Metrica	Risultato Simulativo
Tempo di risposta: $E(T_s)$	1847,54041 +/- 285,99078

Tempo di attesa: $E(T_q)$	1838,54696 +/- 285,99418
Popolazione media: $E(N_s)$	1659,50607 +/- 256,03053
Tempo di interarrivo	1,07771 +/- 0,01055
Popolazione media nella coda: $E(N_q)$	1651,44233 +/- 256,03008

Validazione biglietti per la platea

Metrica	Risultato Simulativo
Tempo di risposta: $E(T_s)$	10,92318 +/- 0,36441
Tempo di attesa: $E(T_q)$	2,10405 +/- 0,14517
Popolazione media: $E(N_s)$	0,76811 +/- 0,03286
Tempo di interarrivo	13,72784 +/- 0,44116
Popolazione media nella coda: $E(N_q)$	0,14862 +/- 0,01135

Valori globali per i server di validazione

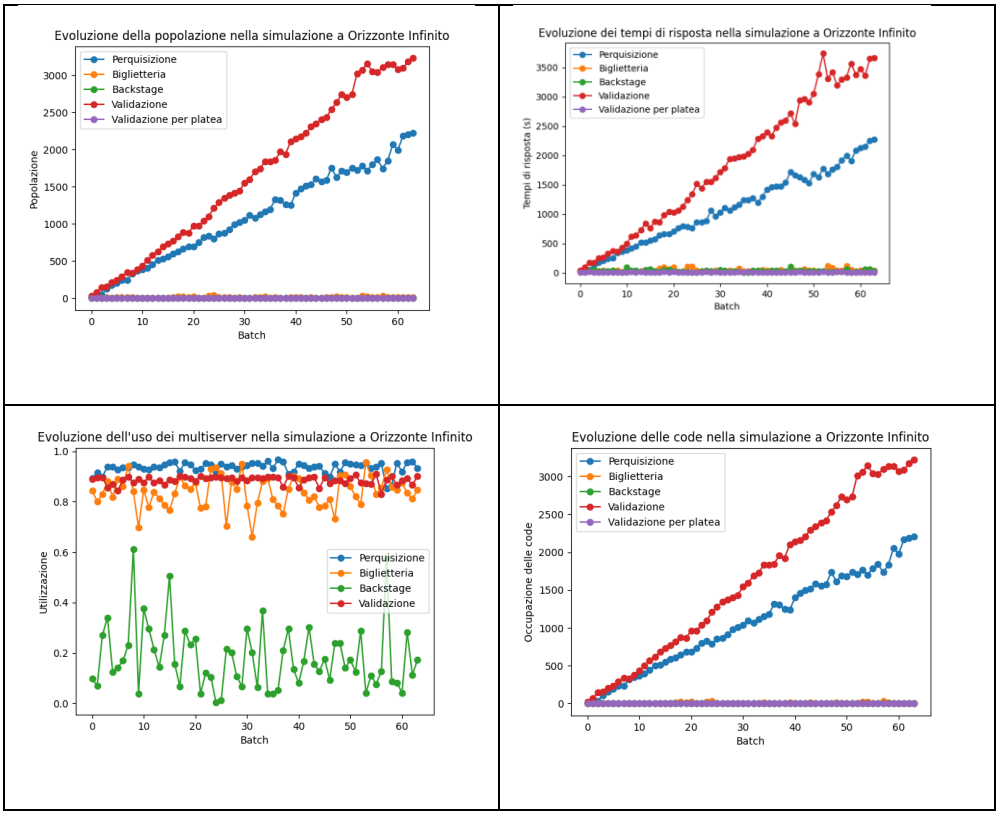
Metrica	Risultato Simulativo
Utilizzo: ρ	0,88556 +/- 0,00393

Tempo di servizio medio: $E(\mathrm{S})$	9,98857 +/- 0,08844

Backstage

Metrica	Risultato Simulativo
Tempo di risposta: $E(T_s)$	30,85646 +/- 4,77235
Tempo di attesa: $E(T_q)$	2,94081 +/- 2,22229
Utilizzo: ρ	0,17987 +/- 0,03173
Popolazione media: $E(N_s)$	0,21108 +/- 0,04648
Tempo di interarrivo	297,91480 +/- 49,25011
Tempo di servizio medio: $E(\mathrm{S})$	27,31522 +/- 3,75688
Popolazione media nella coda: $E(N_q)$	0,03121 +/- 0,02105

Confrontando i valori ottenuti con quelli della simulazione a Orizzonte finito notiamo che mentre alcuni valori restano similari, quelli relativi alle Perquisizioni e alla Validazione dei biglietti non prioritari aumentano notevolmente.



Notiamo dai grafici che l'occupazione di tutte le code tranne il backstage si avvicina ad 1 e che la popolazione, l'occupazione delle code ed i tempi di risposta dei nodi di Validazione e Perquisizione tendono a divergere. Questo comportamento può essere evitato aumentando il numero di serventi disponibili per i due nodi.