

PROGETTO PMCSN

Tribuna Monte Mario
Stadio Olimpico di Roma

Bauco Simone – Esposito Luca Saverio – Staccone Simone

1. Obiettivi
 - 1.1. Descrizione del sistema
 - 1.2. Problematiche
 - 1.3. Obiettivi
2. Modello concettuale
 - 2.1. Introduzione
 - 2.2. Utenti
 - 2.3. Stato
 - 2.4. Eventi
3. Modello delle specifiche
 - 3.1. Distribuzioni utilizzate
4. Modello computazionale
 - 4.1. Linguaggio di programmazione e framework utilizzati
 - 4.2. Eventi
5. Verifica
 - 5.1. Introduzione
 - 5.2. Analisi del sistema
6. Validazione
7. Progettazione degli esperimenti
 - 7.1. Analisi del collo di bottiglia
8. Simulazioni e analisi dei risultati
 - 8.1. Simulazione a orizzonte infinito
 - 8.2. Simulazione a orizzonte finito
9. Modello migliorativo
 - 9.1. Modello concettuale
 - 9.2. Modello delle specifiche
 - 9.3. Modello computazionale
 - 9.4. Verifica
 - 9.5. Validazione
 - 9.6. Progettazione degli esperimenti
 - 9.7. Simulazioni e analisi dei risultati
10. Conclusioni

1. Obiettivi

1.1 Descrizione del sistema

Il sistema analizzato è un settore dello Stadio Olimpico di Roma, in particolare la tribuna Monte Mario. Il settore in questione è il più costoso, viene frequentato generalmente da famiglie. Spesso è possibile trovare anche tifosi ospiti nonostante sia un settore dedicato al tifo di casa. Il settore è suddiviso in varie zone: top nord, nord, top sud, sud, centrale nord, centrale sud, dove il nome ne indica la posizione; per finire la zona premium (VIP), centrale, sopra le panchine, che garantisce la visuale migliore.

L'accesso per tutte le zone, escluso quello premium, è lo stesso, e una volta superati i vari controlli i clienti verranno smistati per raggiungere la zona corretta. I clienti VIP possono essere persone del mondo dello spettacolo, parenti dei calciatori o semplicemente tifosi che vogliono vivere un'esperienza migliore. L'accesso è separato rispetto alle zone normali, per garantire privacy ed un servizio più rapido. Essendo comunque il settore più costoso, per partite "di cartello", ossia quelle più rilevanti, il costo dei biglietti supera anche i 100€ a persona, di conseguenza i clienti si aspettano un servizio efficiente e più rapido rispetto a settori più economici come la curva. Aspetti come i tempi di attesa ma anche la sicurezza percepita sono fondamentali e determinano la qualità offerta.

1.2 Problematiche

Negli ultimi due anni, dopo la riapertura degli stadi a capienza completa, c'è stato un importante incremento dell'affluenza in tutti gli stadi d'Italia. In particolare, nel caso in esame, che riguarda i dati legati alle partite di calcio dell'A.S. Roma, nei 19 match casalinghi della stagione 2022/2023 i biglietti venduti hanno sempre superato i 60.000. In media più dell'88% dello stadio è sempre stato occupato, e questo ha comportato un incremento notevole dei tempi di servizio ed in coda, rispetto all'epoca pre-covid in cui l'affluenza era minore. Il personale, per cercare di ovviare alle problematiche legate all'attesa, non riesce a controllare la totalità dei clienti nelle due perquisizioni all'ingresso, di fatto, "rinunciando" alla verifica degli oggetti immessi nell'impianto. Questo ha comportato una frequenza maggiore di atti di vandalismo all'interno dello stadio come deturpazione delle pareti, lancio di petardi, fumogeni e bombe carta che possono rappresentare un pericolo per la clientela. Gli aspetti cardine presi in considerazione sono stati:

- L'attesa per entrare nello stadio, dovuta principalmente al fatto che gli arrivi si concentrano nell'arco dell'ora antecedente all'evento, generando quindi un collo di bottiglia.
- Il problema delle perquisizioni: spesso, quando il tasso di arrivo cresce, non tutti gli spettatori vengono perquisiti. Ciò è dovuto allo scarso numero di agenti di polizia addetti alle perquisizioni e alla necessità di mantenere i tempi di accesso alla tribuna entro una certa soglia, che permette a tutti di entrare entro l'inizio dell'evento.

La noncuranza di questi aspetti a lungo termine potrebbe portare ad un malcontento e quindi ad una diminuzione delle vendite, portando a perdite nella maggiore fonte d'incassi per la società, oltre a problemi - evidentemente più gravi - legati alla sicurezza.

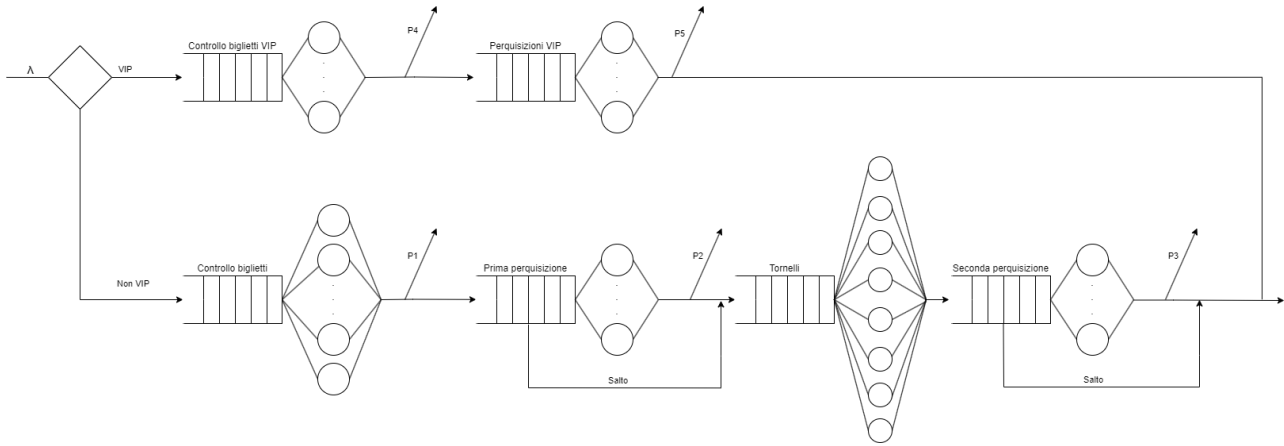
1.3 Obiettivi

- Ridurre il tempo di risposta del sistema.
- Rendere più efficiente il processo di perquisizione, suddividendo gli utenti con zaini/borse da quelli senza, e garantendo che tutte gli utenti siano perquisiti.
- Ridurre i costi: minimizzare il personale, mantenendo tempi di risposta e percentuale di perquisizioni in linea con gli altri obiettivi.

2. Modello concettuale

2.1 Introduzione

Il modello concettuale del sistema in analisi è il seguente:



Dove le probabilità di abbandono del sistema sono:

- P1 -> probabilità di un utente di non superare il controllo biglietti
- P2 -> probabilità di un utente di avere un oggetto pericoloso con sé e di non superare la prima perquisizione
- P3 -> probabilità di un utente di avere un oggetto pericoloso con sé e di non superare la seconda perquisizione
- P4 -> probabilità di un VIP di non superare il controllo biglietti
- P5 -> probabilità di un VIP di avere un oggetto pericoloso con sé e di non superare la perquisizione

2.2 Utenti

Nel sistema, si individuano due tipi di utenti: utenti standard (92.3562% del totale) e utenti VIP (7.6438% del totale), che accedono a parti separate della coda.

2.3 Stato

Lo stato del sistema è dato dallo stato di ogni nodo. In particolare, ogni servente può essere busy o idle, e ogni coda può essere vuota o non vuota.

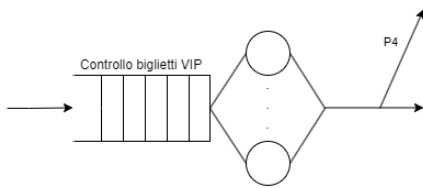
2.4 Eventi

I tipi di eventi sono: arrivo di un nuovo utente, partenza di un utente servito, abbandono a seguito di controllo del biglietto/perquisizione e salto della perquisizione.

In ogni servente, il job viene eseguito immediatamente se la coda è vuota e se il servente non sta eseguendo job, altrimenti il job viene posto in coda, mantenendo una politica di scheduling FIFO.

In seguito, sono descritti nel dettaglio tutti i serventi e i possibili eventi che possono verificarsi al loro interno.

Controllo biglietti VIP

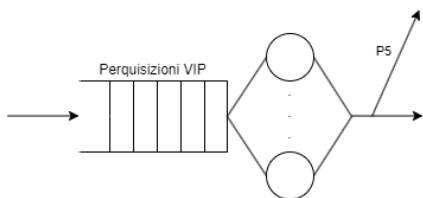


Gli eventi sono:

- Arrivo di un utente VIP
- Uscita di un utente VIP
- Abbandono di un utente (a seguito di biglietto non valido) con probabilità $P4$.

Il modello è un multi-server con coda singola, dato che la coda è unica e sono presenti più steward che controllano l'identità dei possessori dei biglietti.

Perquisizioni VIP

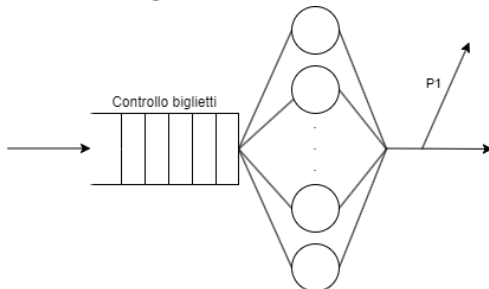


Gli eventi sono:

- Arrivo di un utente VIP
- Uscita di un utente VIP
- Abbandono di un utente (a seguito di perquisizione) con probabilità $P5$.

Il modello è un multi-server con coda singola, dato che la coda per entrare è unica e sono presenti più persone appartenenti alle forze dell'ordine che controllano che nessun oggetto indesiderato venga introdotto nello stadio.

Controllo biglietti

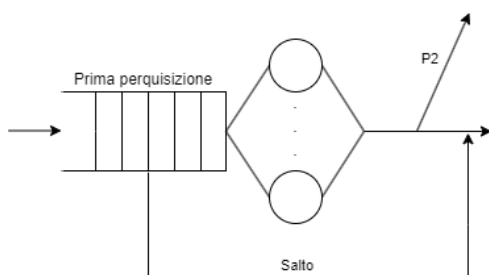


Gli eventi sono:

- Arrivo di un utente
- Uscita di un utente
- Abbandono di un utente (a seguito di biglietto non valido) con probabilità $P1$.

Il modello è un multi-server con coda singola, dato che la coda per entrare è unica e sono presenti più steward che controllano l'identità dei possessori dei biglietti.

Prima perquisizione

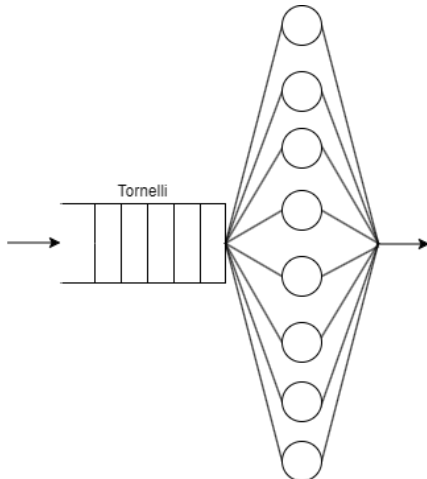


Gli eventi sono:

- Arrivo di un utente
- Uscita di un utente
- Abbandono di un utente (a seguito della perquisizione) con probabilità $P2$
- Salto: passaggio al server successivo (tornelli) con probabilità dipendente dalla coda

Il modello è un multi-server con coda singola, dato che la coda per entrare è unica e sono presenti più persone appartenenti alle forze dell'ordine che controllano che nessun oggetto indesiderato venga introdotto nello stadio.

Tornelli

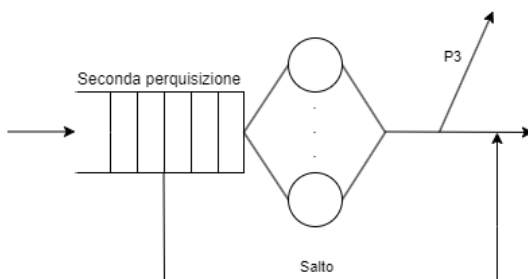


Gli eventi sono:

- Arrivo di un utente
- Uscita di un utente

Il modello è un multi-server con coda singola, dato che la coda per arrivare ai tornelli è unica, ma sono presenti più tornelli per controllare la validità del biglietto.

Seconda perquisizione



Gli eventi sono:

- Arrivo di un utente
- Uscita di un utente
- Abbandono di un utente (a seguito della perquisizione) con probabilità P_3
- Salto: passaggio al servente successivo (ingresso allo stadio) con probabilità dipendente dalla coda.

Il modello è un multi-server con coda singola, dato che la coda per entrare è unica e sono presenti più persone appartenenti alle forze dell'ordine che controllano che nessun oggetto indesiderato venga introdotto nello stadio.

3. Modello delle specifiche

3.1 Distribuzioni utilizzate

Per quanto riguarda gli arrivi (che sono stati suddivisi in tre fasce orarie consecutive) abbiamo scelto di utilizzare, nell'ambito della singola fascia oraria, la distribuzione esponenziale, in quanto possiede le seguenti proprietà:

- Arrivi indipendenti tra loro: nel nostro sistema, gli arrivi non si influenzano tra loro.
- Assenza di memoria: nel nostro sistema, l'arrivo di un nuovo utente non è condizionato dagli arrivi degli utenti precedenti.
- Processo di Poisson: gli arrivi seguono un processo di Poisson (eventi casuali e indipendenti nel tempo, frequenza media degli arrivi costante).

L'idea di suddividere gli arrivi in tre fasce orarie, quindi, è particolarmente utile per modellare le fluttuazioni del traffico in arrivo.

Per quanto riguarda i servizi abbiamo scelto di utilizzare la distribuzione esponenziale, per gli stessi motivi esposti precedentemente.

Per quanto riguarda i salti delle perquisizioni, abbiamo utilizzato una funzione di probabilità, realizzata ad hoc, che cresce linearmente rispetto al numero medio di utenti in coda. In particolare, è necessario modellare il salto come un evento necessario per evitare la presenza in coda di troppi utenti, quindi tempi di attesa troppo prolungati. Da esperienze personali, infatti, abbiamo riscontrato che i poliziotti presenti cominciano a lasciar passare le persone in coda, senza alcuna perquisizione, quando queste cominciano ad avvicinarsi e a superare il numero di persone medie previste.

Essendo un fenomeno difficile da modellare, data l'aleatorietà del comportamento dei poliziotti addetti alle perquisizioni e non avendo a disposizione dati empirici relativi alle popolazioni medie in coda, abbiamo deciso di rimandare la definizione della funzione di probabilità successivamente all'acquisizione di dati ottenuti attraverso la simulazione.

Alcuni dati (numero di steward e poliziotti, numero di tornelli, tempi medi di servizio) sono stati richiesti ed ottenuti direttamente da dipendenti dello Stadio Olimpico, altri dati (e.g. capacità della tribuna, numero medio di spettatori (VIP e non)) sono dati ufficiali. Altri dati, ad esempio riguardanti le probabilità di perquisizione fallita, non sono direttamente disponibili, quindi sono stati ragionevolmente stimati.

La capienza della tribuna Monte Mario è di 14266 posti, di cui 1013 VIP. Nella stagione 2022/2023, l'affluenza massima è stata del 91.01282%; quindi, ipotizzando che gli spettatori fossero uniformemente suddivisi in tutti i settori, consideriamo 922 spettatori VIP e 12062 spettatori non VIP.

La proporzione di utenti VIP è pari al 7.6438% del totale ($q = 0.076438$).

- **Controllo biglietti:** $M|M|10$ con $E(S) = 7$ s
- **Prima perquisizione:** $M|M|20$ con $E(S) = 20$ s
- **Tornelli:** $M|M|8$ con $E(S) = 5$ s
- **Seconda perquisizione:** $M|M|20$ con $E(S) = 20$ s
- **Controllo biglietti VIP:** $M|M|4$ con $E(S) = 7$ s
- **Perquisizioni VIP:** $M|M|5$ con $E(S) = 20$ s

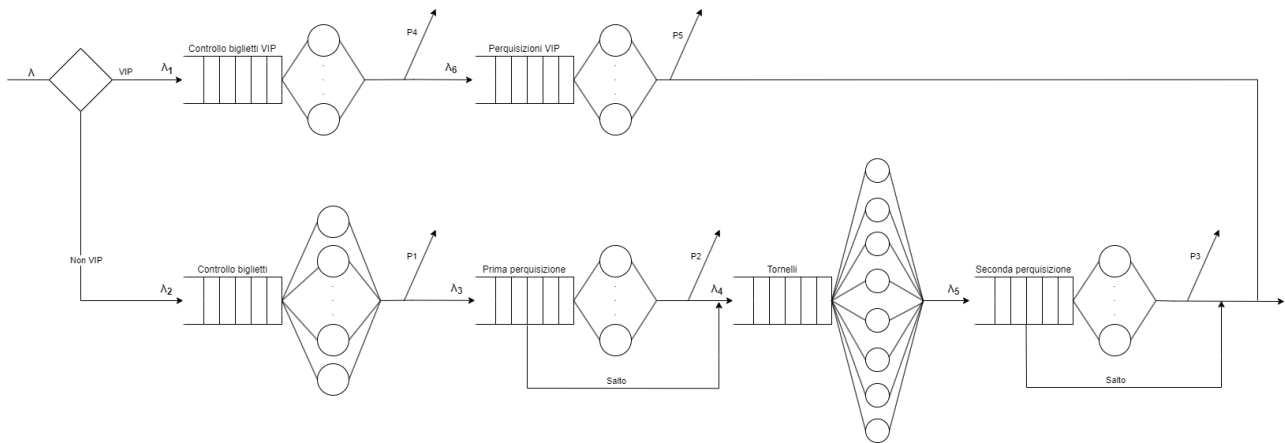
Il costo medio di uno steward è di 55 €/evento, mentre il costo medio di un poliziotto è di 80 €/evento.

A seguito di osservazioni (date da esperienze personali), si è visto che:

- Per quanto riguarda il controllo biglietti (vip e non), questo non va a buon fine se:
 - Il biglietto non è intestato all'utente
 - Il biglietto non è valido per il settore considerato
 - Il biglietto è falso
 - Non viene presentato un documento valido

In media, 8 persone su 100 sono bloccate al controllo biglietti (P1 = 0.08), mentre 10 persone su 100 sono bloccate al controllo biglietti VIP (P4 = 0.1).

- Per quanto riguarda la prima e seconda perquisizione, queste non vanno a buon fine se l'utente perquisito possiede oggetti non ammessi. In media, ciò accade per 2 persone su 100 (P2 = P3 = 0.02).
- Per quanto riguarda la perquisizione VIP, la probabilità di espulsione è dell'1% (P5 = 0.01).



Le equazioni di traffico sono:

$$\lambda_1 = q \cdot \lambda$$

$$\lambda_2 = (1 - q) \cdot \lambda$$

$$\lambda_3 = (1 - P1) \cdot \lambda_2$$

$$\lambda_4 = \lambda_3 \cdot (1 - P2) \cdot (1 - P_{salto}(n)) + \lambda_3 \cdot P_{salto}(n)^1$$

$$\lambda_5 = \lambda_4$$

$$\lambda_6 = (1 - P4) \cdot \lambda_1$$

¹ $P_{salto}(n)$ rappresenta la probabilità di salto, che è dipendente dal numero n di persone in coda

La matrice di routing è la seguente:

/	EST	CB	PP	T	SP	CBV	PV	ING
EST	0	$1 - q$	0	0	0	q	0	0
CB	p_1	0	$(1 - p_1)$	0	0	0	0	0
PP	$(1 - p_{salto}(n))p_2$	0	0	$(1 - p_{salto}(n))(1 - p_2) + p_{salto}(n)$	0	0	0	0
T	0	0	0	0	1	0	0	0
SP	$(1 - p_{salto}(n))p_3$	0	0	0	0	0	0	$(1 - p_{salto}(n))(1 - p_3) + p_{salto}(n)$
CBV	p_4	0	0	0	0	0	$1 - p_4$	0
PV	p_5	0	0	0	0	0	0	$1 - p_5$

EST = Esterno, CB = Controllo Biglietti, PP = Prima Perquisizione, T = Tornelli, CBV = Controllo Biglietti VIP, PV = Perquisizione VIP, ING = Ingresso allo stadio

3.2 Descrizione degli arrivi

Dalla nostra esperienza, abbiamo visto che la maggioranza degli arrivi (70%), per una partita che inizia alle 21:00, si hanno nella fascia oraria 19:30 - 20:30, mentre si hanno più arrivi prima delle 19:30 (20%) rispetto che dopo le 20:30 (10%).

Consideriamo, in totale, 12984 utenti (12062 utenti normali e 922 utenti VIP).

Di conseguenza, il tasso di arrivo globale è di 1.2022 persone/s, che viene ripartito nelle tre fasce orarie come segue:

- 18:30 - 19:30: 20%
- 19:30 - 20:30: 70 %
- 20:30 - 21:30: 10%

ottenendo i seguenti tassi di arrivo:

- 18:30 - 19:30: 0.24044 persone/s
- 19:30 - 20:30: 0.84154 persone/s
- 20:30 - 21:30: 0.12022 persone/s

4. Modello computazionale

4.1 Linguaggio di programmazione e framework utilizzati

Nell'ambito di questo progetto, è stato utilizzato il linguaggio di programmazione Java, ritenuto flessibile e adatto alle nostre esigenze. Inoltre, la nostra maggiore capacità e dimestichezza con questo linguaggio, ci ha spinto ad utilizzarlo.

4.2 Eventi

Il codice relativo alla simulazione è contenuto nelle classi `ComputationalModel.java` e `VIPComputationalModel.java`, relative alla simulazione rispettivamente della coda non-vip e della coda vip. Queste, infatti, nella simulazione sono state considerate indipendenti tra loro, essendo fissa la percentuale di un utente di essere oppure no un utente VIP.

In seguito, viene descritta nel dettaglio la sola classe `ComputationalModel.java`, in quanto la classe `VIPComputationalModel.java` è del tutto analoga.

All'interno della classe, per mantenere traccia degli eventi, viene utilizzato un array di oggetti di tipo `MsqEvent`, così definito:

```
MsqEvent[] events = new MsqEvent[
    ALL_EVENTS_TICKET +
    ALL_EVENTS_FIRST_PERQUISITION +
    ALL_EVENTS_TURNSTILES +
    ALL_EVENTS_SECOND_PERQUISITION];
```

Questo rappresenta l'insieme degli eventi possibili nell'ingresso standard, in particolare:

INDICE	Tipologia evento
events [0]	Arrivo al servizio di controllo biglietti.
events [1-10]	Completamento del controllo biglietto da parte di un operatore.
events [11]	Abbandono dal controllo biglietti a causa di incongruenze nel biglietto o altri motivi.
events [12]	Arrivo al servizio di prima perquisizione.
events [13-32]	Completamento della prima perquisizione da parte di un agente.
events [33]	Salto della perquisizione dovuto a congestione eccessiva.
events [34]	Abbandono dalla prima perquisizione dovuto ad esito negativo.
events [35]	Arrivo al servizio dei tornelli.
events [36-43]	Completamento dell'obliterazione del biglietto nel tornello.
events [44]	Arrivo al servizio di seconda perquisizione.
events [45-64]	Completamento della seconda perquisizione da parte di un agente.
events [65]	Salto della perquisizione dovuto a congestione eccessiva.
events [66]	Abbandono dalla seconda perquisizione dovuto ad esito negativo.

In dettaglio, l'oggetto `MsqEvent` ha la seguente forma:

```
class MsqEvent {
    double t;           /* next event time */
    int x;              /* event status, 0 or 1 */
}
```

dove `t` indica l'istante (in termini di tempo di clock) del prossimo evento di quel tipo, mentre `x` indica se l'evento è attivo oppure no: l'evento viene eseguito al tempo indicato da `t` se e solo se `x` vale 1. Altre strutture di supporto alla simulazione sono le seguenti:

- ```

class MsqT {
 double current; /* current time */
 double next; /* next (most imminent) event time*/
}

```

Simula il flusso temporale, implementando il clock di simulazione. In ogni istante, in `current` viene mantenuto il valore attuale del clock, mentre in `next` il valore di clock dell'evento successivo.

- ```

class MsqSum {
    double service;           /* accumulated sums of */
    long served;              /* service times */
}

```

Mantiene, in ogni istante e per ogni nodo del sistema, il numero di job processati e il tempo di processamento totale.

La simulazione per la coda non-VIP, dunque, avviene nel metodo `main()` della classe `ComputationalModel.java`, che esegue un ciclo fintanto che non vale una delle seguenti condizioni:

- `events[0].x == 0`: l'evento di arrivo al primo servente del sistema è disattivato;
- `numberTicketCheck + numberFirstPerquisition + numberTurnstiles + numberSecondPerquisition == 0`: il numero di job in ogni nodo del sistema è pari a 0.

All'interno di ogni ciclo, viene invocata la funzione `int nextEvent(MsqEvent[] event)` che determina, dato l'array degli eventi, l'evento successivo (cioè, l'evento attivo più prossimo nel tempo). Dato l'indice dell'evento successivo, questo viene gestito nella *subroutine* corrispondente generando, ove occorre, l'evento successivo a quello appena gestito.

Per garantire la randomicità degli eventi, è stato utilizzato un generatore di Lehmer, implementato nella classe `Rngs.java`, al fine di generare tempi di arrivo e tempi di servizio. In particolare, in ogni simulazione è stato utilizzato un unico generatore, applicando un approccio multi-stream: ovvero, è stato scelto uno stream diverso per ogni processo stocastico (tempi di interarrivo, tempi di servizio, abbandoni e salti per ogni nodo), in modo da garantire un disaccoppiamento tra i numeri random generati per ogni processo.

Infatti, utilizzando lo stesso oggetto `Rngs r = new Rngs();` abbiamo invocato una sola volta la funzione `r.plantseed()` e, successivamente, abbiamo selezionato uno stream differente per ogni evento aleatorio attraverso la funzione `r.selectStream(...)` cercando di scegliere un indice il più possibile distante dall'ultimo precedentemente usato e dal successivo successivamente usato. Per questo abbiamo partizionato uniformemente i 256 possibili stream forniti dalla libreria, in base al numero di invocazioni della funzione `r.random()`.

4.3 Generazione dei salti delle perquisizioni

Come già accennato in precedenza, i salti delle perquisizioni sono generati in funzione della popolazione media in coda. Realizzando delle simulazioni a orizzonte finito, effettuando in particolare 150 replicazioni, abbiamo ottenuto dati riguardanti le popolazioni medie in coda. Tramite la funzione Python `find_outliers_IQR()`, riportata in figura, abbiamo determinato il lower bound e l'upper bound nei dati,

tramite il metodo degli IQR. Ovvero ci siamo basati sul primo e sul terzo quartile del dataset così generato per ottenere una retta che rappresenta la crescita della probabilità da 0% a 80%.

```

3 def find_outliers_IQR(data, threshold=1.5):
4     # Calcola il primo e il terzo quartile
5     Q1, Q3 = np.percentile(data, [25, 75])
6     # Calcola l'IQR
7     IQR = Q3 - Q1
8     # Calcola la soglia inferiore e superiore
9     lower_bound = Q1 - (threshold * IQR)
10    upper_bound = Q3 + (threshold * IQR)
11    return lower_bound, upper_bound

```

Figura 1: funzione `find_outliers_IQR()`, presente nello script `outliers.py`

La funzione di probabilità degli skip è stata determinata, quindi, nel seguente modo:

$$\frac{y - 80}{80 - 0} = \frac{x - upper_bound}{upper_bound - lower_bound}$$

Da cui ottengo la retta:

$$y = 80 \frac{x - upper_bound}{upper_bound - lower_bound} + 80$$

$$y = \frac{80}{upper_bound - lower_bound}x + \left(80 - \frac{upper_bound}{upper_bound - lower_bound}\right)$$

Quindi, utilizzando questa relazione, abbiamo valutato la percentuale come $\min(0.8, y/100)$ nei casi in cui $y > 0$, mentre $y = 0$ nei casi in cui $y < 0$.

$$Prob(skip) = \begin{cases} \min\left(0.8, \frac{y}{100}\right) & y \geq 0 \\ 0 & y < 0 \end{cases}$$

In questo modo abbiamo una percentuale che cresce linearmente quando il numero di persone nel server supera il lower bound, ovvero una distanza dalla media pari ad 1.5 volte la distanza tra media e primo quartile, fino ad avere il massimo nel lower bound che corrisponde ad una probabilità pari all'80%.



Figura 2: grafico della funzione di probabilità degli skip ottenuta partendo dal dataset ottenuto da finite horizon simulation

5. Verifica

5.1 Introduzione

Nel processo di verifica, è stata utilizzata la seguente configurazione di sistema:

- Controllo biglietti: 10 server
- Prima perquisizione: 20 server
- Tornelli: 8 server
- Seconda perquisizione: 20 server
- Controllo biglietti VIP: 4 server
- Perquisizione VIP: 5 server

considerando la prima fascia oraria (18:30 – 19:30).

L'impiego di questa configurazione ha consentito di ridurre i tempi di attesa in coda nei nodi *Prima perquisizione* e *Seconda perquisizione*, in modo da eliminare il fenomeno dei “salti” delle perquisizioni, rendendo il sistema studiabile dal punto di vista analitico, al fine di verificare i risultati ottenuti dalla simulazione.

5.2 Analisi del sistema

Controllo biglietti VIP

Sono dati il numero di server $m = 4$, il tempo di servizio medio di ogni server $E(S_i) = 7$ s e il tasso di arrivo $\lambda_{VTC} = 0.05122$ utenti/s.

$$\mu_i = \frac{1}{E(S_i)} = 0.14286 \text{ utenti/s}$$

$$\rho = \frac{\lambda_{VTC}}{m\mu_i} = 0.08963$$

$$E(T_Q) = \frac{P_Q E(S)}{1 - \rho} = \frac{(mp)^m}{m! (1 - \rho)} \cdot \frac{1}{\sum_{i=0}^{m-1} \frac{(mp)^i}{i!} + \frac{(mp)^m}{m! (1 - \rho)}} \cdot \frac{E(S)}{1 - \rho} = 0.00102 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S_i) = 7.00102 \text{ s}$$

$$E(\text{interarrivi}) = \frac{1}{\lambda_{VTC}} = 19.52286 \text{ s}$$

$$E(N_S) = \lambda_{VTC} E(T_Q) + \rho m = 0.35857 \text{ utenti}$$

Perquisizioni VIP

Sono dati il numero di server $m = 5$, il tempo di servizio medio di ogni server $E(S_i) = 20$ s e il tasso di arrivo $\lambda_{VP} = (1 - P_4) \cdot \lambda_{VTC} = 0.0461$ utenti/s.

$$\mu_i = \frac{1}{E(S_i)} = 0.05 \text{ utenti/s}$$

$$\rho = \frac{\lambda_{VP}}{m\mu_i} = 0.1844$$

$$E(T_Q) = \frac{P_Q E(S)}{1 - \rho} = \frac{(m\rho)^m}{m!(1 - \rho)} \cdot \frac{1}{\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1 - \rho)}} \cdot \frac{E(S)}{1 - \rho} = 0.01328 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S_i) = 20.01328 \text{ s}$$

$$E(\text{interarrivi}) = \frac{1}{\lambda_{VP}} = 21.692 \text{ s}$$

$$E(N_S) = \lambda_{VP} E(T_Q) + \rho m = 0.92261 \text{ utenti}$$

Controllo biglietti

Sono dati il numero di server $m = 10$, il tempo di servizio medio di ogni server $E(S_i) = 7 \text{ s}$ e il tasso di arrivo $\lambda_{TC} = 0.67011 \text{ utenti/s}$.

$$\mu_i = \frac{1}{E(S_i)} = 0.05 \text{ utenti/s}$$

$$\rho = \frac{\lambda_{TC}}{m\mu_i} = 0.46908$$

$$E(T_Q) = \frac{P_Q E(S)}{1 - \rho} = \frac{(m\rho)^m}{m!(1 - \rho)} \cdot \frac{1}{\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1 - \rho)}} \cdot \frac{E(S)}{1 - \rho} = 0.03232 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S_i) = 7.03232 \text{ s}$$

$$E(\text{interarrivi}) = \frac{1}{\lambda_{TC}} = 1.49229 \text{ s}$$

$$E(N_S) = \lambda_{TC} E(T_Q) + \rho m = 4.71246 \text{ utenti}$$

Prima perquisizione

Sono dati il numero di server $m = 20$, il tempo di servizio medio di ogni server $E(S_i) = 20 \text{ s}$ e il tasso di arrivo $\lambda_{PP} = \lambda_{TC}(1 - P_1) = 0.6165 \text{ utenti/s}$.

$$\mu_i = \frac{1}{E(S_i)} = 0.03333 \text{ utenti/s}$$

$$\rho = \frac{\lambda_{PP}}{m\mu_i} = 0.6165$$

$$E(T_Q) = \frac{P_Q E(S)}{1 - \rho} = \frac{(m\rho)^m}{m!(1 - \rho)} \cdot \frac{1}{\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1 - \rho)}} \cdot \frac{E(S)}{1 - \rho} = 0.08109 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S_i) = 20.08109 \text{ s}$$

$$E(\text{interarrivi}) = \frac{1}{\lambda_{PP}} = 1.62206 \text{ s}$$

$$E(N_S) = \lambda_{PP}E(T_Q) + \rho m = 12.37999 \text{ utenti}$$

Tornelli

Sono dati il numero di server $m = 8$, il tempo di servizio medio di ogni server $E(S_i) = 5 \text{ s}$ e il tasso di arrivo $\lambda_T = \lambda_{PP}(1 - P_2) = 0.60417 \text{ utenti/s}$.

$$\mu_i = \frac{1}{E(S_i)} = 0.2 \text{ utenti/s}$$

$$\rho = \frac{\lambda_T}{m\mu_i} = 0.37761$$

$$E(T_Q) = \frac{P_Q E(S)}{1 - \rho} = \frac{(m\rho)^m}{m!(1 - \rho)} \cdot \frac{1}{\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1 - \rho)}} \cdot \frac{E(S)}{1 - \rho} = 0.01352 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S_i) = 5.01352 \text{ s}$$

$$E(\text{interarrivi}) = \frac{1}{\lambda_T} = 1.65516 \text{ s}$$

$$E(N_S) = \lambda_T E(T_Q) + \rho m = 3.02905 \text{ utenti}$$

Seconda perquisizione

Sono dati il numero di server $m = 20$, il tempo di servizio medio di ogni server $E(S_i) = 20 \text{ s}$ e il tasso di arrivo $\lambda_{SP} = \lambda_T = 0.60417 \text{ utenti/s}$.

$$\mu_i = \frac{1}{E(S_i)} = 0.03333 \text{ utenti/s}$$

$$\rho = \frac{\lambda_{SP}}{m\mu_i} = 0.60417$$

$$E(T_Q) = \frac{P_Q E(S)}{1 - \rho} = \frac{(m\rho)^m}{m!(1 - \rho)} \cdot \frac{1}{\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1 - \rho)}} \cdot \frac{E(S)}{1 - \rho} = 0.0651 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S_i) = 20.0651 \text{ s}$$

$$E(\text{interarrivi}) = \frac{1}{\lambda_{SP}} = 1.65516 \text{ s}$$

$$E(N_S) = \lambda_{SP} E(T_Q) + \rho m = 12.12273 \text{ utenti}$$

Per la fase di verifica, sono state prodotte due simulazioni a orizzonte infinito tramite *Batch Means*:

- Per la coda standard: $b = 7168$, $k = 128$.
- Per la coda VIP: $b = 1024$, $k = 50$.

La scelta della dimensione del batch, cioè del valore del parametro b , è stata guidata dal criterio di *Banks et al. [2001]*², secondo cui il valore di b deve aumentare fintanto che il valore di lag-autocorrelation, tra le medie dei batch, non scende sotto 0.2. Ciò è stato verificato tramite l'analisi dei dati di output implementata nella classe Acs.java.

Le seguenti tabelle confrontano i valori analitici con i valori dati dalla simulazione, per ogni nodo del sistema.

Statistica	Risultato analitico	Risultato simulazione
$E(T_S)$	7.00102 s	7.02114 +/- 0.06626
$E(T_Q)$	0.00102 s	0.00246 +/- 0.00160
ρ	0.08963	0.08972 +/- 0.00131
tempo di interarrivo	19.52286 s	19.58569 +/- 0.16380
$E(S)$	7 s	7.02043 +/- 0.06620 s
$E(N_S)$	0.35857	0.35892 +/- 0.00522

Tabella 1: Controllo biglietti VIP

Statistica	Risultato analitico	Risultato simulazione
$E(T_S)$	20.01328 s	19.99114 +/- 0.17999
$E(T_Q)$	0.01328 s	0.01394 +/- 0.01055
ρ	0.1844	0.18383 +/- 0.00231
tempo di interarrivo	21.692 s	21.75430 +/- 0.18721
$E(S)$	20 s	19.97672 +/- 0.17827
$E(N_S)$	0.92261	0.91981 +/- 0.01166

Tabella 2: Perquisizioni VIP

Statistica	Risultato analitico	Risultato simulazione
$E(T_S)$	7.03232 s	7.02447 +/- 0.01493 s
$E(T_Q)$	0.03232 s	0.03067 +/- 0.00222 s
ρ	0.46908	0.46935 +/- 0.00131
tempo di interarrivo	1.49299 s	1.49030 +/- 0.00294 s
$E(S)$	7 s	6.99379 +/- 0.01410 s
$E(N_S)$	4.71246	4.71408 +/- 0.01374

Tabella 3: Controllo biglietti

² Banks, Carson, Nelson, and Nicol (2001, page 438) recommend that the batch size be increased until the lag one autocorrelation between batch means is less than 0.2.

Statistica	Risultato analitico	Risultato simulazione
$E(T_S)$	20.08109 s	20.07040 +/- 0.04609 s
$E(T_Q)$	0.08109 s	0.08017 +/- 0.00709 s
ρ	0.6165	0.61718 +/- 0.00189
tempo di interarrivo	1.62206 s	1.61971 +/- 0.00332 s
$E(S)$	20 s	19.98995 +/- 0.04408 s
$E(N_S)$	12.37999	12.39322 +/- 0.03965

Tabella 4: Prima perquisizione

Statistica	Risultato analitico	Risultato simulazione
$E(T_S)$	5.01352 s	5.00973 +/- 0.01090 s
$E(T_Q)$	0.01352 s	0.01299 +/- 0.00107 s
ρ	0.37761	0.37796 +/- 0.00118
tempo di interarrivo	1.65516 s	1.65281 +/- 0.00348 s
$E(S)$	5 s	4.99674 +/- 0.01064 s
$E(N_S)$	3.02905	3.03151 +/- 0.00964

Tabella 5: Tornelli

Statistica	Risultato analitico	Risultato simulazione
$E(T_S)$	20.0651 s	20.09710 +/- 0.04329 s
$E(T_Q)$	0.0651 s	0.06978 +/- 0.00752 s
ρ	0.60417	0.60595 +/- 0.00193
tempo di interarrivo	1.65516 s	1.65284 +/- 0.00350 s
$E(S)$	20 s	20.02705 +/- 0.04143
$E(N_S)$	12.12273	12.16128 +/- 0.04016

Tabella 6: Seconda perquisizione

6. Validazione

Pur avendo contattato l'azienda, non è stato possibile reperire dati storici riguardanti il sistema di interesse.

Di conseguenza, al fine di validare il sistema, sono stati effettuati i seguenti controlli di consistenza:

- Al crescere del tasso di arrivo (e.g. passando dalla fascia oraria 18:30 – 19:30 alla più congestionata fascia oraria 19:30 – 20:30), si ha un aumento dei tempi di servizio, in ogni servente.
- Al crescere del tasso di arrivo, si ha un aumento nei salti alle perquisizioni.

Per realizzare i controlli di consistenza, sono state utilizzate le simulazioni a orizzonte infinito introdotte nella verifica, ottenendo i seguenti risultati:

- Per il primo controllo di consistenza, sono stati confrontati i tempi di risposta medi ottenuti su tutti i serventi tramite *Batch Means*, considerando una volta la fascia oraria 18:30-19:30 e una volta la fascia oraria 19:30-20:30.

La configurazione di server è la seguente:

- Controllo biglietti: 47 server
- Prima perquisizione: 65 server
- Tornelli: 12 server
- Seconda perquisizione: 65 server
- Controllo biglietti VIP: 4 server
- Perquisizione VIP: 5 server

Nodo del sistema	$E(T_s)_{18:30-19:30}$	$E(T_s)_{19:30-20:30}$
Controllo biglietti	19.98227 s	174.11762 s
Prima perquisizione	29.98355 s	111.63701 s
Tornelli	4.99683 s	7.37716 s
Seconda perquisizione	30.04131 s	90.09165 s
Controllo biglietti VIP	7.02114 s	7.13912 s
Perquisizione VIP	19.98519 s	23.22043 s

Tabella 7: Confronto tra i tempi di risposta medi nelle prime due fasce orarie

- Per il secondo controllo di consistenza, sono stati confrontati il numero medio di salti nei nodi *Prima perquisizione* e *Seconda perquisizione*, considerando una volta la fascia oraria 18:30-19:30 e una volta la fascia oraria 19:30-20:30. In seguito, sono riportati i risultati ottenuti da una simulazione a orizzonte infinito, tramite Batch Means ($b = 7168$; $k = 128$), utilizzando la configurazione (10, 20, 8, 20).

Nodo del sistema	Numero di skip (18:30-19:30)	Numero di skip (19:30-20:30)
Prima perquisizione	0.00000 +/- 0.00000	4009.05469 +/- 32.10326
Seconda perquisizione	0.00000 +/- 0.00000	37.35156 +/- 38.48490

Tabella 8: Confronto tra il numero di salti nelle prime due fasce orarie

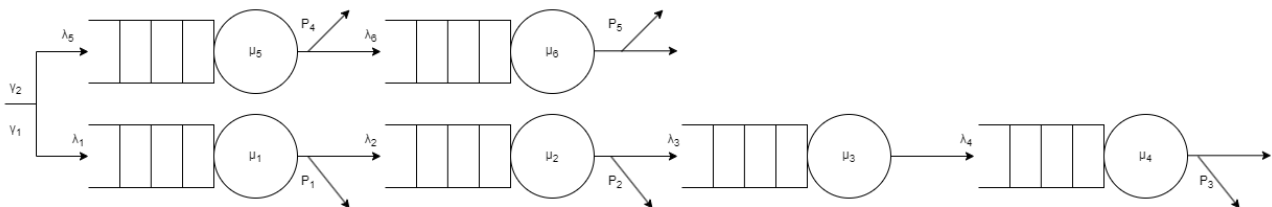
7. Progettazione degli esperimenti

7.1 Introduzione

Sono stati progettati e realizzati i seguenti esperimenti:

- Analisi del collo di bottiglia, al fine di individuare nel sistema il nodo che rappresenta il *bottleneck*;
- Simulazioni a orizzonte infinito, tramite il metodo *Batch Means*, allo scopo di analizzare il sistema nell'ambito delle singole fasce orarie, per determinare configurazioni in grado di mantenere prestazioni stazionarie in un contesto di esecuzione all'infinito;
- Simulazioni a orizzonte finito, tramite replicazioni, per analizzare le prestazioni del sistema considerando l'effettivo contesto di esecuzione (cioè, le tre fasce orarie), allo scopo di determinare diverse configurazioni di server in grado di raggiungere obiettivi diversi.

7.2 Analisi del collo di bottiglia



Servizio medio dei centri del sistema

$$S1 = 1/\mu_1 = 7 \text{ s}$$

$$S2 = 1/\mu_2 = 20 \text{ s}$$

$$S3 = 1/\mu_3 = 5 \text{ s}$$

$$S4 = 1/\mu_4 = 20 \text{ s}$$

$$S5 = 1/\mu_5 = 7 \text{ s}$$

$$S6 = 1/\mu_6 = 20 \text{ s}$$

Calcolo del numero medio di visite ai centri del sistema

Per prima cosa, è stato calcolato il flusso entrante in ogni nodo al pari delle perdite dovute ai controlli:

$$\lambda_1 = \gamma_1$$

$$\lambda_2 = \gamma_1(1-P_1)$$

$$\lambda_3 = \lambda_2(1-P_2)$$

$$\lambda_4 = \lambda_3$$

$$\lambda_5 = \gamma_2$$

$$\lambda_6 = \gamma_2(1-P_4)$$

Ottenendo:

$$\lambda_1 = \gamma_1$$

$$\lambda_2 = \gamma_1(1-P_1)$$

$$\lambda_3 = \gamma_1(1-P_1)(1-P_2)$$

$$\lambda_4 = \gamma_1(1-P_1)(1-P_2)$$

$$\lambda_5 = \gamma_2$$

$$\lambda_6 = \gamma_2(1-P_4)$$

Da cui:

$$v_1 = \lambda_1 / \gamma_1 = 1$$

$$v_2 = \lambda_2 / \gamma_1 = 1 - P_1 = 0,92$$

$$v_3 = \lambda_3 / \gamma_1 = (1 - P_1) (1 - P_2) = 0,9016$$

$$v_4 = \lambda_4 / \gamma_1 = (1 - P_1) (1 - P_2) = 0,9016$$

$$v_5 = \lambda_5 / \gamma_2 = 1$$

$$v_6 = \lambda_6 / \gamma_2 = 1 - P_4 = 0,92$$

Calcolo della **domanda media per i centri del sistema**

$$D_1 = v_1 S_1 = 7s$$

$$D_2 = v_2 S_2 = (1 - P_1) * 20 = 0,92 * 20 = 18,4 s$$

$$D_3 = v_3 S_3 = (1 - P_1) (1 - P_2) * 5 = 0,92 * 0,98 * 5 = 4,508 s$$

$$D_4 = v_4 S_4 = (1 - P_1) (1 - P_2) * 20 = 0,92 * 0,98 * 20 = 18,032 s$$

$$D_5 = v_5 S_5 = 7 s$$

$$D_6 = v_6 S_6 = (1 - P_4) * 20 s = 18,4 s$$

D1	7 s
D2	18.4 s
D3	4.508 s
D4	18.032 s
D5	7 s
D6	18.4 s

Tabella 9: Domande di servizio in ogni centro

Successivamente si è effettuato lo stesso calcolo utilizzando il numero di visite ottenute attraverso una simulazione a orizzonte finito per validare l'evidenza analitica.

Nodo del sistema	Numero di ingressi
Controllo biglietti Vip	922.91333 +/- 4.83448
Perquisizione Vip	830.93333 +/- 4.64272
Controllo biglietti	12058.35333 +/- 16.92985
Prima perquisizione	11093.94000 +/- 17.27554
Tornelli	10963.66000 +/- 17.64587
Seconda perquisizione	10963.65333 +/- 17.64518

Tabella 10: Ingressi ad ogni centro, ottenuti tramite 150 replicazioni sulla configurazione (4, 5, 10, 20, 8, 20)

Nodo del sistema	Numero medio di visite	Tempo medio di servizio
Controllo biglietti Vip	1.00000 +/- 0.00000	6.99211 +/- 0.03857
Perquisizione Vip	0.90034 +/- 0.00097	20.02869 +/- 0.11507
Controllo biglietti	1.00000 +/- 0.00000	6.99929 +/- 0.01119
Prima perquisizione	0.92002 +/- 0.00027	20.00113 +/- 0.03230
Tornelli	0.90922 +/- 0.00027	5.00631 +/- 0.00774
Seconda perquisizione	0.90922 +/- 0.00027	19.99771 +/- 0.03251

Tabella 11: Numero medio di visite e tempi medi di servizio in ogni centro

Nodo del sistema	Domanda di servizio
Controllo biglietti Vip	6.99211 +/- 0.03857
Perquisizione Vip	18.03260 +/- 0.12303
Controllo biglietti	6.99929 +/- 0.01119
Prima perquisizione	18.40140 +/- 0.03517
Tornelli	4.55184 +/- 0.00084
Seconda perquisizione	18.18230 +/- 0.03496

Tabella 12: Domande di servizio in ogni centro

Quindi si può notare come il secondo nodo di entrambe le code (VIP e utenti normali) sia il collo di bottiglia del sistema. In particolare, nel caso di utenti standard, il *bottleneck* corrisponde alla prima perquisizione, mentre nel caso VIP all'unica perquisizione. Infatti, sia dai risultati analitici che da quelli ottenuti attraverso la simulazione ad orizzonte finito, si nota come la maggiore domanda di servizio arrivi da questi nodi.

8. Simulazioni e analisi dei risultati

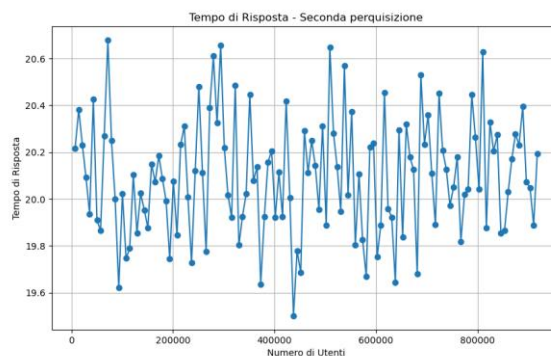
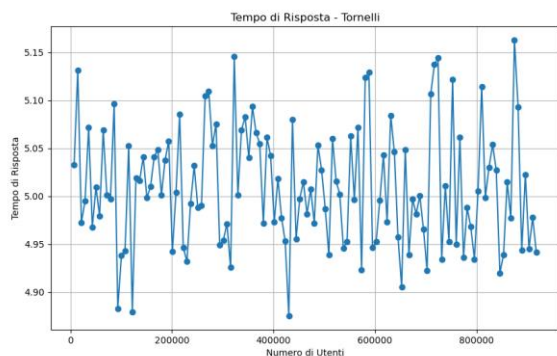
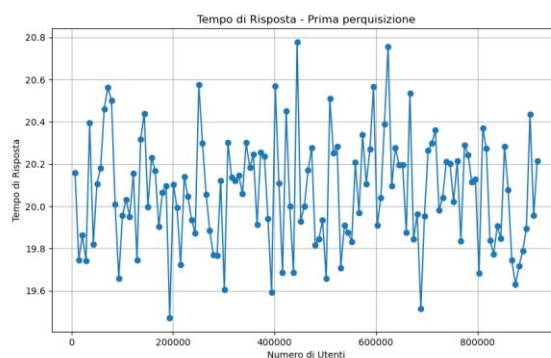
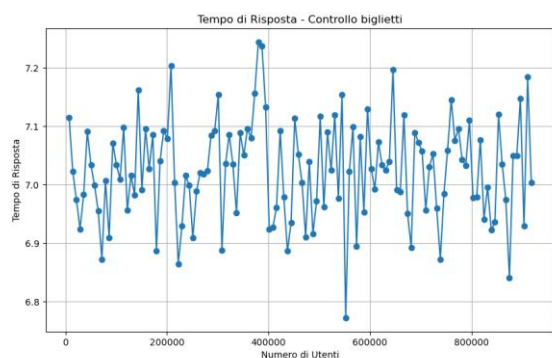
I seguenti esperimenti si limitano al caso della coda standard, che è stata ritenuta cruciale dal punto di vista dell'ottimizzazione dell'intero sistema. Infatti, la coda Vip presenta già ottime prestazioni³, quindi non è stato considerato fondamentale studiare a fondo le sue capacità tramite simulazioni a orizzonte finito/infinito.

8.1 Simulazione a orizzonte infinito

Il secondo esperimento riguarda l'utilizzo di una simulazione a orizzonte infinito (*Batch Means*) per trovare il minimo numero di server necessari in ogni nodo, in ogni fascia oraria.

Fascia oraria 18:30-19:30

Utilizzando la configurazione di server (10, 20, 8, 20) per la coda standard, da una simulazione *Batch Means* si ottengono i seguenti risultati:



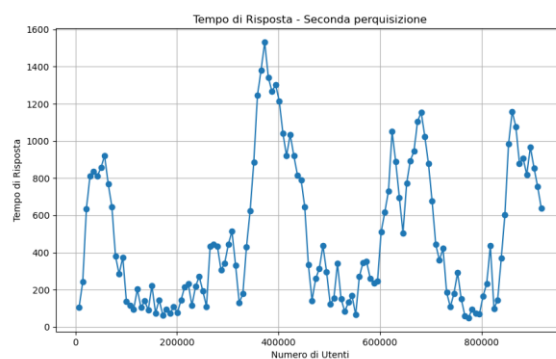
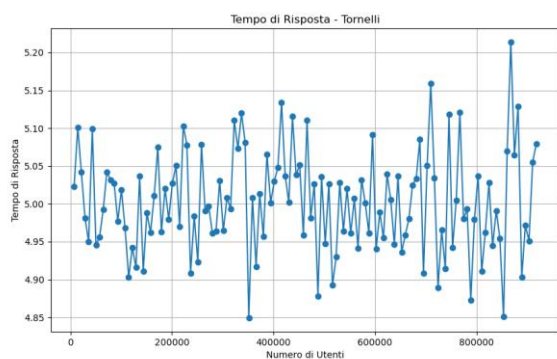
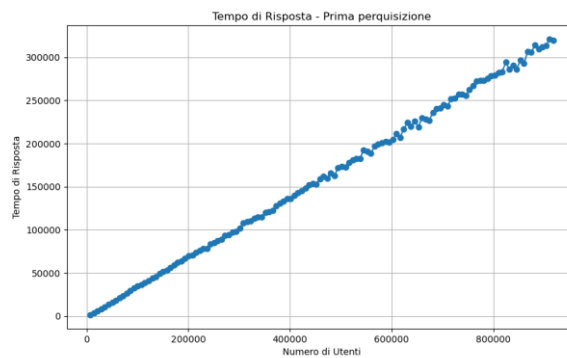
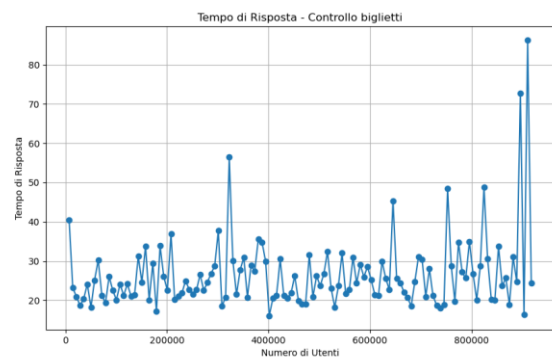
Il tempo medio di accesso allo stadio, nella prima fascia oraria, è di 52.2017 secondi⁴.

Dimezzando il numero di server, cioè, passando alla configurazione (5, 10, 8, 10)⁵, si ottengono i seguenti risultati:

³ In particolare, il tempo medio di accesso allo stadio, ottenuto da una simulazione a orizzonte finito con 150 replicazioni, è pari a 29.46179 secondi.

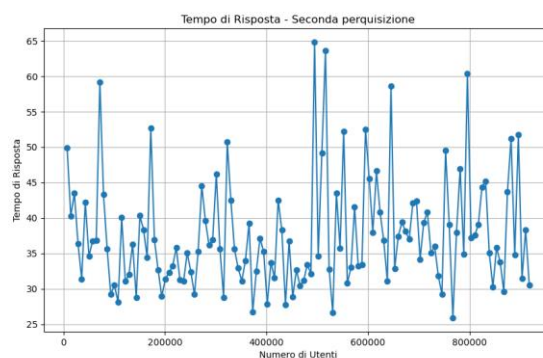
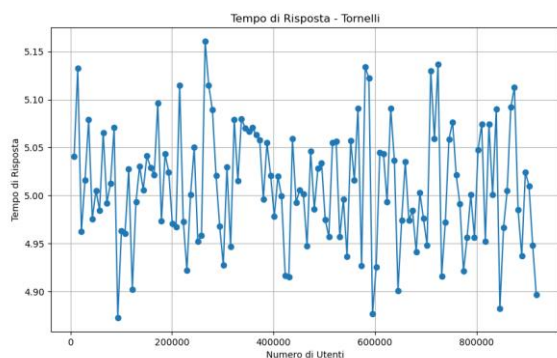
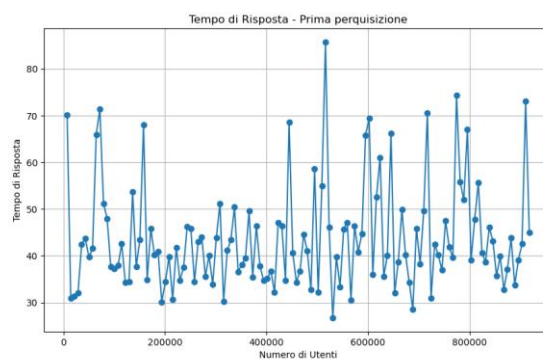
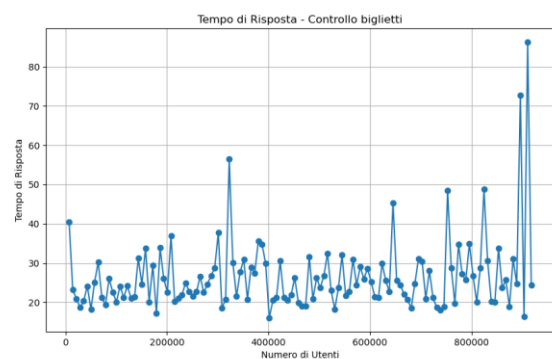
⁴ Questo risultato è stato ottenuto considerando la somma dei tempi medi di risposta in ogni server.

⁵ Si noti che non ha senso diminuire il numero di tornelli, in quanto la riduzione di questi non porta a risparmio di risorse economiche.



Come si vede dai grafici, questa configurazione porta ad un'esplosione dei tempi di risposta del nodo *Prima perquisizione*, che diventa un collo di bottiglia per l'intero sistema.

Adottando la configurazione (5, 13, 8, 13), si ottengono i seguenti risultati:

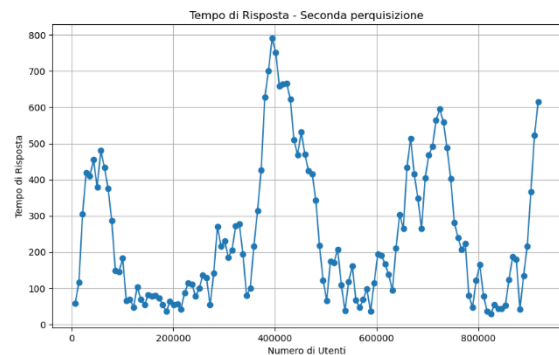
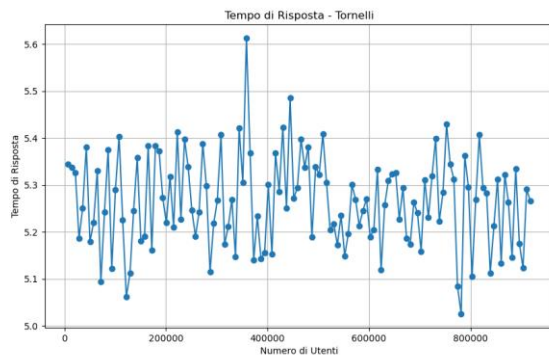
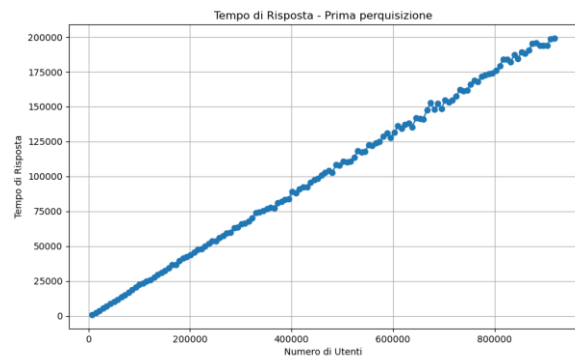
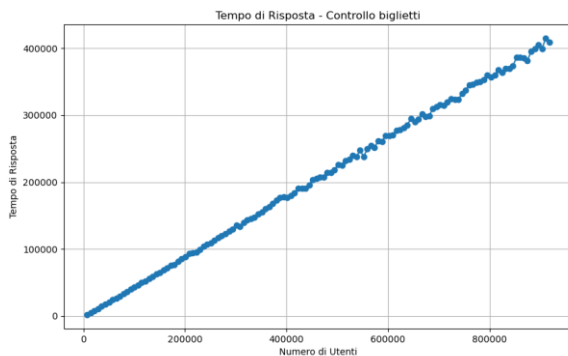


Alla configurazione (5, 13, 8, 13) corrisponde un tempo medio di ingresso allo stadio di 112.77748 secondi.

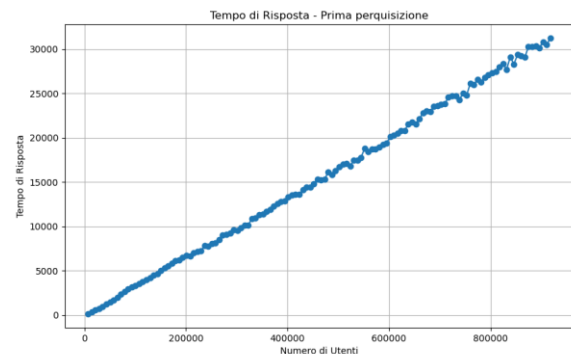
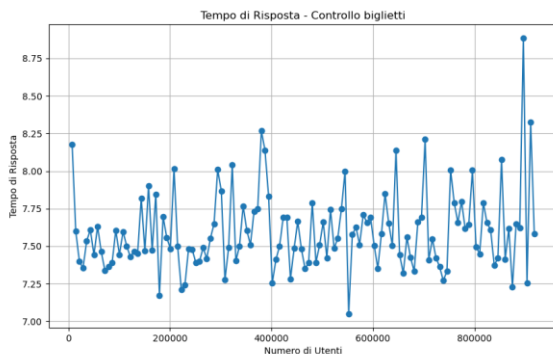
Di conseguenza, adottare la configurazione (5, 13, 8, 13) porta ad un aumento del tempo medio di ingresso allo stadio di 60.57578 secondi, a fronte però di un risparmio di 5 steward e 14 poliziotti per la prima fascia oraria.

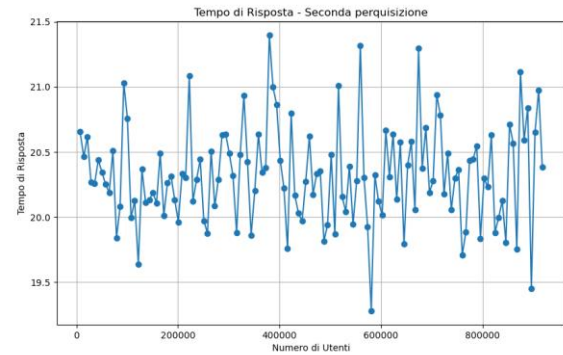
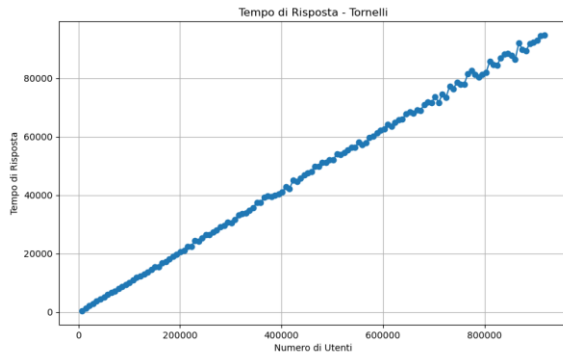
Fascia oraria 19:30-20:30

Questa è la fascia oraria più congestionata, di conseguenza ci aspettiamo che la configurazione standard, cioè (10, 20, 8, 20), provochi un'esplosione nei tempi di risposta. Infatti, considerando la configurazione standard si ha:



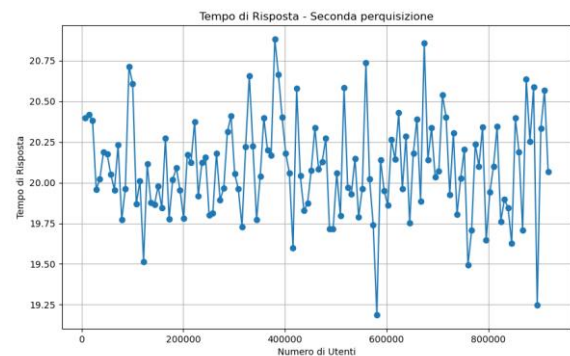
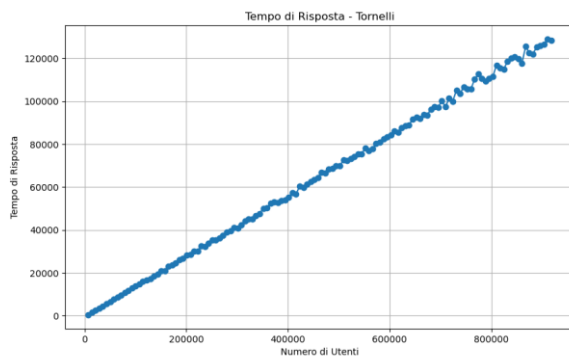
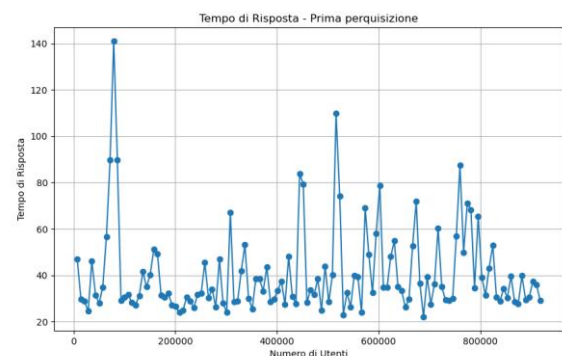
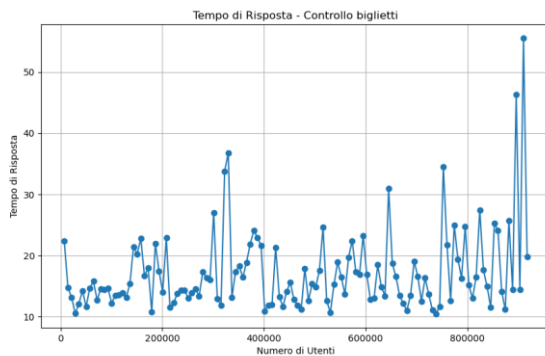
Raddoppiando il numero di serventi, quindi passando alla configurazione (20, 40, 8, 40), si ha:





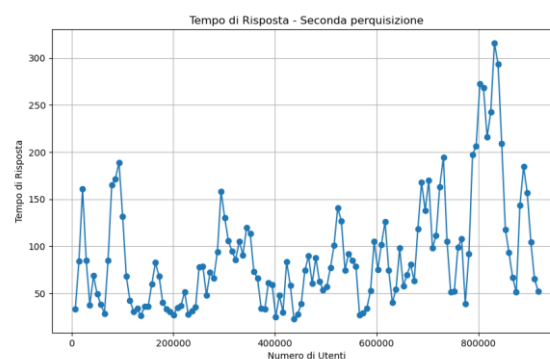
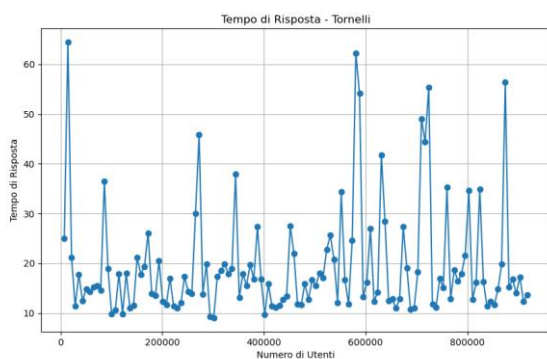
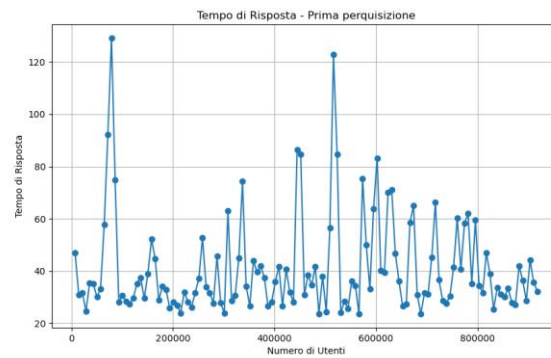
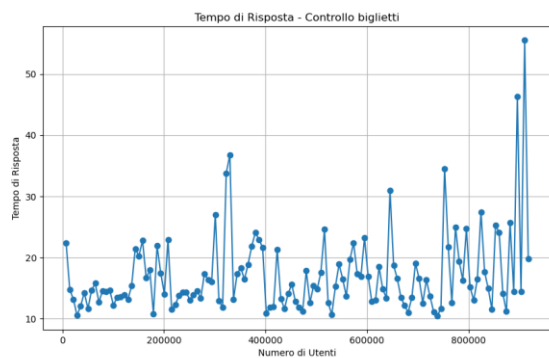
Dai risultati, si vede che questa configurazione porta a tempi di risposta stazionari per il nodo *Controllo biglietti*, tuttavia non è ancora sufficiente per il nodo *Prima Perquisizione*, per cui i tempi di risposta crescono all'infinito.

La configurazione (17, 44, 8, 43) produce i seguenti risultati:



Si ha quindi una condizione stazionaria per il nodo *Prima perquisizione* (quindi anche del nodo *Seconda perquisizione*), mentre non si ha stazionarietà per il nodo *Tornelli*. Essendo i tornelli un'infrastruttura fissa del sistema, non è possibile modificare la quantità di questi; tuttavia, la simulazione a orizzonte infinito produce dei risultati che sono validi quando il sistema è operativo in un tempo infinito, mentre il nostro sistema opera per un tempo finito (3 ore). Di conseguenza, è plausibile che il nodo *Tornelli* possa rimanere in una condizione stazionaria senza alcuna modifica.

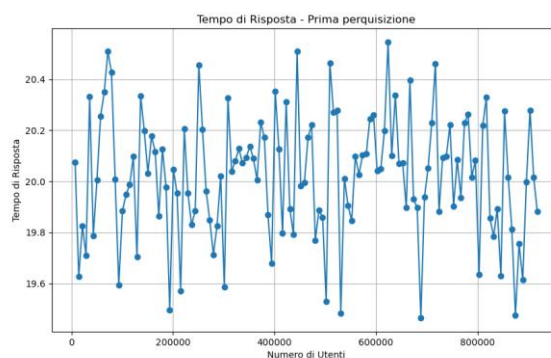
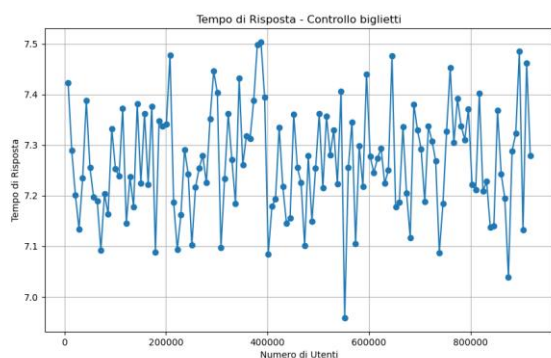
Tuttavia, all'unico scopo di misurare i tempi di risposta, è stato reso stazionario anche il nodo *Tornelli*, con la configurazione (17, 44, 11, 43), che produce i seguenti risultati:

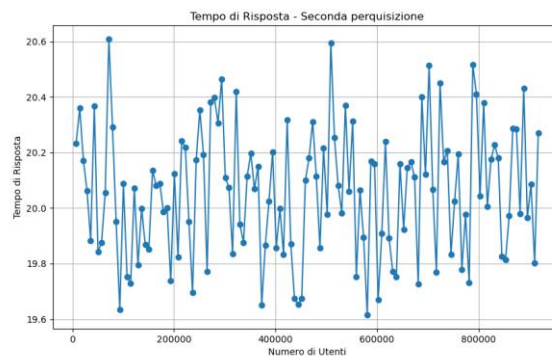
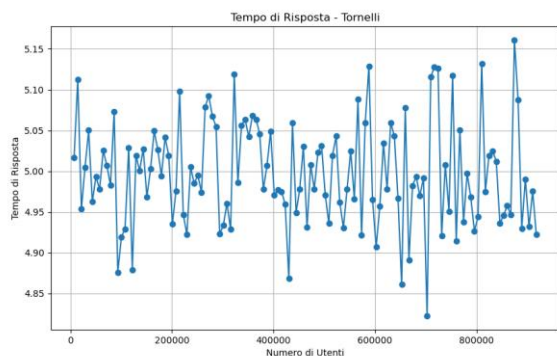


Il tempo medio di accesso allo stadio, con questa configurazione, è di circa 122.21424 s.

Fascia oraria 20:30-21:30

In questa fascia oraria si ha il 10% degli arrivi totali, di conseguenza ha senso effettuare una simulazione a orizzonte infinito considerando la configurazione ottimale per la prima fascia oraria, in cui si hanno il 20% degli arrivi, cioè (5, 13, 8, 13). Di seguito sono presentati i risultati con questa configurazione:

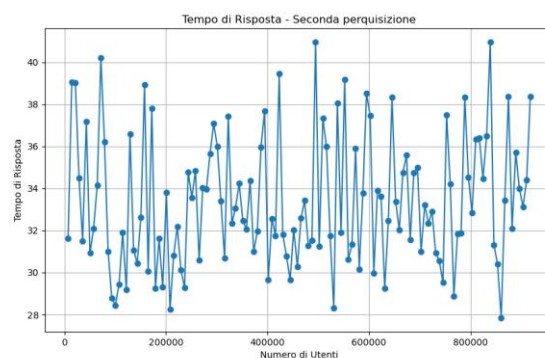
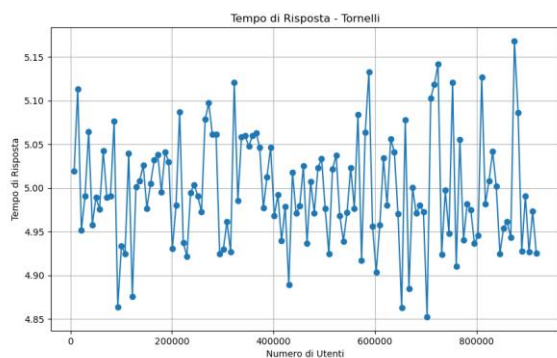
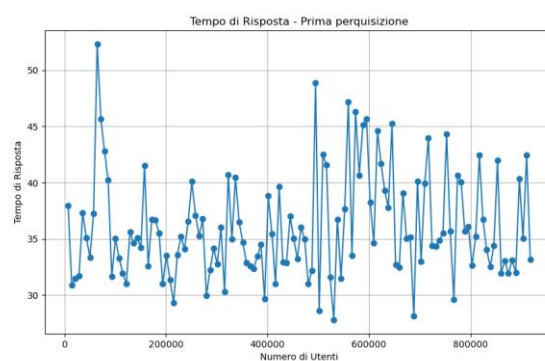
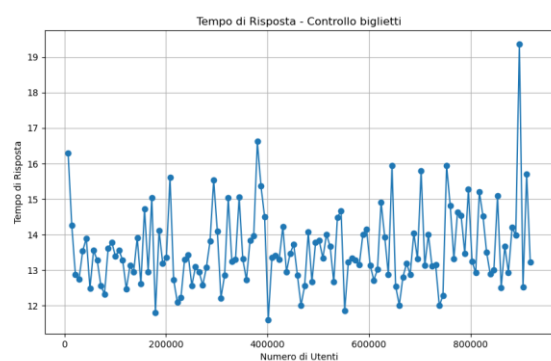




Utilizzando questa configurazione, il tempo medio di ingresso allo stadio è di 52.34373 secondi.

Inoltre, in tutti i nodi il tempo di coda è circa 0 secondi, di conseguenza si può ridurre il numero di server.

La configurazione (3, 7, 8, 7) dà i seguenti risultati:



Il tempo medio di accesso allo stadio, in questo caso, è di 88.01371 secondi.

Conclusioni

Le tre configurazioni “ottimali”, corrispondenti alle tre fasce orarie, sono:

- 18:30-19:30: (5, 13, 8, 13)
- 19:30-20:30: (17, 44, 11, 43)
- 20:30-21:30: (3, 7, 8, 7)

In complesso, nell’arco delle tre ore, vengono impiegati 25 ore di lavoro per gli steward e 127 ore di lavoro per i poliziotti, mentre nella configurazione standard si hanno 30 ore di lavoro per gli steward e 120 ore di lavoro per i poliziotti.

Queste configurazioni mantengono il sistema stabile in un contesto di esecuzione all'infinito, e presentano tempi di risposta medi inferiori a tre minuti, inoltre garantiscono che tutti gli utenti siano perquisiti (cioè, si hanno sempre zero skip nelle perquisizioni). Di conseguenza, l'utilizzo della sola configurazione della seconda fascia oraria, ad esempio, garantisce tempi di risposta inferiori a tre minuti e zero skip in tutte le fasce orarie.

8.2 Simulazione a orizzonte finito

Inizialmente, è stata effettuata una simulazione a orizzonte finito sul sistema utilizzando la configurazione standard, cioè (10, 20, 8, 20), al fine di determinare il tempo medio di risposta totale. Di seguito sono presentati i risultati.

Nodo del sistema	$E(T_S)$	$E(T_Q)$	ρ
Controllo biglietti	911.29143 +/- 4.01766 s	904.29110 +/- 4.01467 s	0.78142 +/- 0.00091
Prima perquisizione	863.32950 +/- 4.92952 s	843.32660 +/- 4.91969 s	0.88491 +/- 0.00053
Tornelli	5.21584 +/- 0.00675 s	0.21828 +/- 0.00339 s	0.54869 +/- 0.00067
Seconda perquisizione	65.55268 +/- 2.61003 s	45.56339 +/- 2.60151 s	0.87785 +/- 0.00100

Tabella 13: Risultati da 150 replicazioni con la configurazione (10, 20, 8, 20)

Il tempo medio di risposta totale è di 1845.38945 secondi, cioè circa 31 minuti.

Lo scopo del secondo esperimento è di utilizzare la simulazione a orizzonte finito per trovare tre configurazioni che rispettano, rispettivamente, i seguenti tre requisiti di QoS:

- Tempo di ingresso allo stadio minore di 20 minuti. ($E(T_S) < 20 \text{ min}$)
- Minimizzare il numero di personale mantenendo la percentuale di utenti perquisiti superiore al 75% tra tutti gli utenti che entrano nello stadio (quindi mantenendo il 75% di perquisiti al secondo controllo).
- Mantenere il numero medio di utenti per nodo inferiore alle 500 persone, per garantire una maggiore sicurezza nel processo di ingresso allo stadio. ($E(N_S) < 500 \text{ utenti}$)

Effettuando una simulazione a orizzonte finito con la configurazione standard ($n = 150$), si ottengono i seguenti risultati:

Nodo del sistema	$E(T_S)$	Numero di salti	ρ
Controllo biglietti	911.08234 +/- 7.27602 s	/	0.78218 +/- 0.0165
Prima perquisizione	862.41645 +/- 8.56298 s	4590.26667 +/- 33.90826	0.88505 +/- 0.00099
Tornelli	5.20629 +/- 0.01235 s	/	0.54592 +/- 0.00123
Seconda perquisizione	56.68934 +/- 3.29031 s	0.00000 +/- 0.00000	0.87356 +/- 0.00177

Tabella 14: Risultati da 150 replicazioni con la configurazione (10, 20, 8, 20)

Quindi, come già visto, con la configurazione (10, 20, 8, 20) si ha tempo di risposta medio totale di 1835.39442 secondi, cioè circa 31 minuti, e percentuale di utenti perquisiti pari al 57.716% nel nodo *Prima perquisizione* e pari al 100% nel nodo *Seconda perquisizione*.

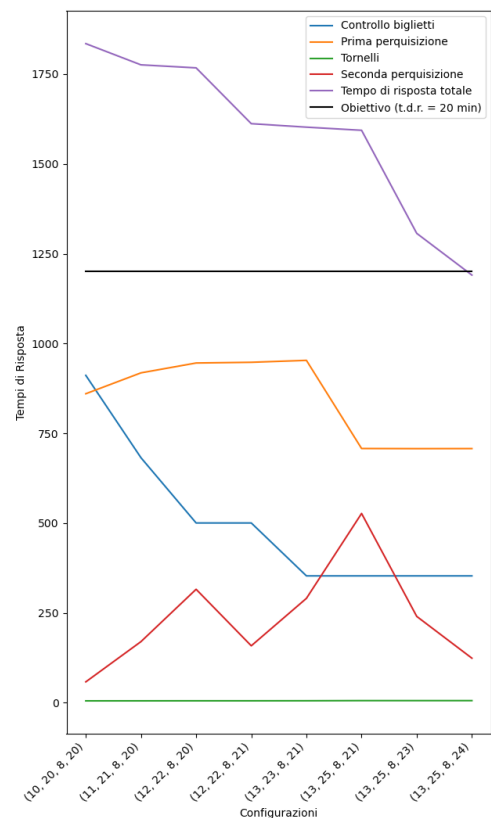
Per raggiungere il primo obiettivo, dunque, risulta evidente che bisogna aumentare il numero di serventi nei nodi *Controllo biglietti*, *Prima perquisizione* e *Seconda perquisizione* (il tempo di risposta relativo a quest'ultimo servente è basso, rispetto al nodo *Prima perquisizione*, perché quest'ultimo fa da collo di bottiglia per il sistema, in particolare per i serventi successivi).

In particolare, sono state effettuate simulazioni ad orizzonte finito con le seguenti configurazioni:

- (10, 20, 8, 20)
- (11, 21, 8, 20)
- (12, 22, 8, 20)
- (12, 22, 8, 21)
- (13, 23, 8, 21)
- (13, 25, 8, 21)
- (13, 25, 8, 23)
- (13, 25, 8, 24)

In figura, sono riportati i risultati ottenuti adottando le configurazioni di serventi di cui sopra.

La configurazione (13, 25, 8, 24) raggiunge l'obiettivo.



Per raggiungere il secondo obiettivo possiamo diminuire il numero di poliziotti adibiti alle perquisizioni. In particolare, solamente diminuendo il numero di serventi nel secondo nodo di perquisizione si ottengono delle prestazioni accettabili e che raggiungono l'obiettivo preposto. Infatti, con la configurazione (10,20,8,17) si ottiene che all'incirca il 75,63% di chi entra nello stadio viene perquisito, con un tempo di risposta medio del sistema di circa 40 min.

Diminuendo anche nella prima perquisizione a 17 serventi, quindi con la configurazione (10,17,8,17), si ottiene un tempo di risposta pari a 41 min, con la totalità degli utenti che vengono perquisiti nel secondo nodo di perquisizioni.

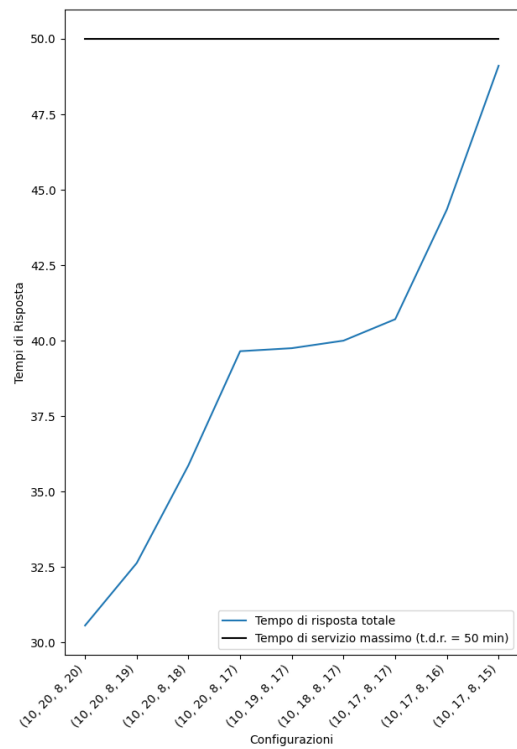
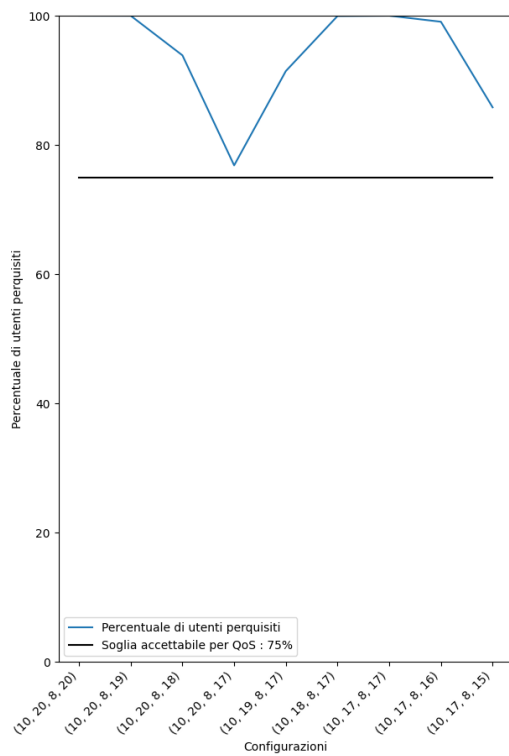
Analogamente quindi, diminuendo ancora i serventi solo nel secondo nodo, ottengo la configurazione (10,17,8,15), che avendo ben 8 poliziotti in meno rispetto alla configurazione iniziale ha una percentuale di perquisizioni pari al 85,8%, con un tempo di risposta medio che però risulta di 49 min.

In generale quindi, nel secondo nodo di perquisizione possono esserci meno serventi di quelli presenti nel primo nodo. Con i risultati ottenuti si capisce che si può far crescere quanto si vuole il tempo di risposta medio del sistema mantenendo più del 75% delle perquisizioni, dato che questo obiettivo dipende dalla proporzione tra server nel primo nodo e nel secondo nodo di perquisizione. Realisticamente però, avere un tempo di accesso allo stadio superiore ai 50 min in media, non renderebbe possibile l'accesso in tempo per l'inizio della partita anche a chi si presenta con il giusto anticipo. Quindi non tenendo conto del tempo di risposta, la configurazione ottimale per minimizzare i costi è (10,17,8,15).

In particolare, i grafici mostrano tempo di risposta e percentuale di utenti perquisiti in base alle seguenti configurazioni:

- (10,20,8,20)

- (10,20,8,19)
- (10,20,8,18)
- (10,20,8,17)
- (10,19,8,17)
- (10,18,8,17)
- (10,17,8,17)
- (10,17,8,16)
- (10,17,8,15)



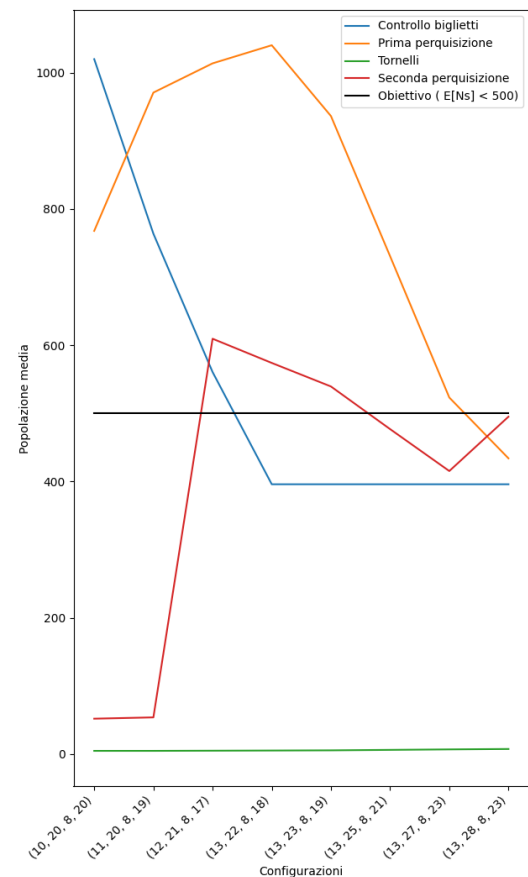
Per raggiungere il terzo obiettivo, si può notare subito che, con la configurazione standard, già al ticket check si ottengono un numero elevato di utenti nel nodo (all'incirca 1020), quindi per garantire una diminuzione di questo parametro è stata necessaria una redistribuzione del personale.

Proprio per questo, visti i risultati ottenuti dai precedenti esperimenti, si è proceduto diminuendo il numero di serventi prima nel secondo nodo di perquisizione e successivamente nel primo, in modo da mantenere il più possibile numeri bassi per gli utenti in coda di questi due nodi.

Per raggiungere questo obiettivo si è dovuto aumentare il numero di poliziotti e steward presenti, ottenendo come configurazione ottimale (13,28,8,23)

In particolare, i grafici mostrano tempo di risposta e percentuale di utenti perquisiti in base alle seguenti configurazioni:

- (10,20,8,20)
- (11,20,8,19)
- (12,21,8,17)
- (13,22,8,18)
- (13,23,8,19)
- (13,25,8,21)
- (13,27,8,23)
- (13,28,8,23)



9. Modello migliorativo

9.1 Introduzione

A valle dei risultati ottenuti dagli esperimenti nelle sezioni precedenti, è stato ritenuto opportuno modificare il sistema al fine di includere una nuova tipologia di utenti, cioè gli abbonati⁶, e di soddisfare i seguenti requisiti di QoS:

- Tempo di accesso allo stadio per gli utenti standard abbonati inferiore a 15 minuti.
- Tempo di accesso allo stadio per gli utenti standard non abbonati inferiore a 25 minuti.
- Tempo di accesso allo stadio per gli utenti VIP inferiore a 3 minuti.
- Percentuale di utenti perquisita pari al 100%.

pur minimizzando il numero di serventi impiegati.

9.2 Modello concettuale

Sono state introdotte le seguenti modifiche:

- Unire i due nodi di perquisizione in un unico nodo, dopo il nodo *Tornelli*.
Dall'analisi del *bottleneck* implementata nella sezione precedente, si vede che il nodo *Prima perquisizione* agisce da collo di bottiglia per il sistema; di conseguenza, utilizzando la configurazione standard, si vede che in questo nodo viene perquisito solamente circa il 60% del totale, mentre nel nodo *Seconda perquisizione* il tasso di perquisizione è del 100%. Dunque, abbiamo ritenuto opportuno rimuovere il nodo *Prima perquisizione* e mantenere il solo nodo *Seconda perquisizione*. La scelta di posizionare l'unico nodo di perquisizione dopo il nodo *Tornelli* discende da considerazioni sulla sicurezza: perquisire gli utenti prima dell'obliterazione del biglietto (cioè, prima dei tornelli) potrebbe consentire ad un utente che non supera la perquisizione di ritentare quest'ultima e, seppur con una probabilità molto bassa, di saltarla, vista la presenza del fenomeno degli *skip*. Ponendo invece il nodo *Perquisizione* dopo il nodo *Tornelli*, se un utente non supera la perquisizione, questo non può accedere di nuovo allo stadio, in quanto il biglietto può essere obliterato una sola volta.
- Aggiungere classi di priorità *size-based* alle perquisizioni.
In particolare, gli utenti che hanno un tempo di servizio minore di 10 secondi, appartengono alla prima classe di priorità, mentre gli altri appartengono alla seconda.
Questo raffinamento deriva dall'individuazione di due categorie di utenti, cioè *utenti con zaino* e *utenti senza zaino*, che richiedono tempi diversi per la perquisizione. La scelta di introdurre *size-based* minimizza il tempo di attesa degli utenti senza zaino, che richiedono un minor tempo di perquisizione.
- Aggiungere una coda dedicata per gli abbonati.
Nel modello migliorativo è stata individuata una terza categoria di utenti, cioè gli utenti che possiedono un abbonamento. Per questa tipologia di utenza, è stato scelto di realizzare una coda dedicata, per quanto riguarda i nodi *Tornelli* e *Perquisizione*, al fine di garantire a questa categoria un tempo di accesso allo stadio ridotto, rispetto al caso standard.
- Unire i serventi *Controllo biglietti VIP* e *Perquisizione VIP* in un unico servente, aggiungendo code di priorità *size-based*.

⁶ Gli abbonati sono il 10% degli utenti standard.

Vista la quantità ridotta di utenti VIP, rispetto a utenti standard, è stata considerata la possibilità di accorpare i nodi di controllo biglietti e perquisizione in un unico nodo, in cui vengono effettuate entrambe le operazioni, al fine di ridurre il tempo di ingresso allo stadio anche per questa categoria. Analogamente al caso delle perquisizioni per utenti standard, sono state introdotte due classi di priorità size-based.

Dunque, nel sistema sono presenti tre tipi di utenti:

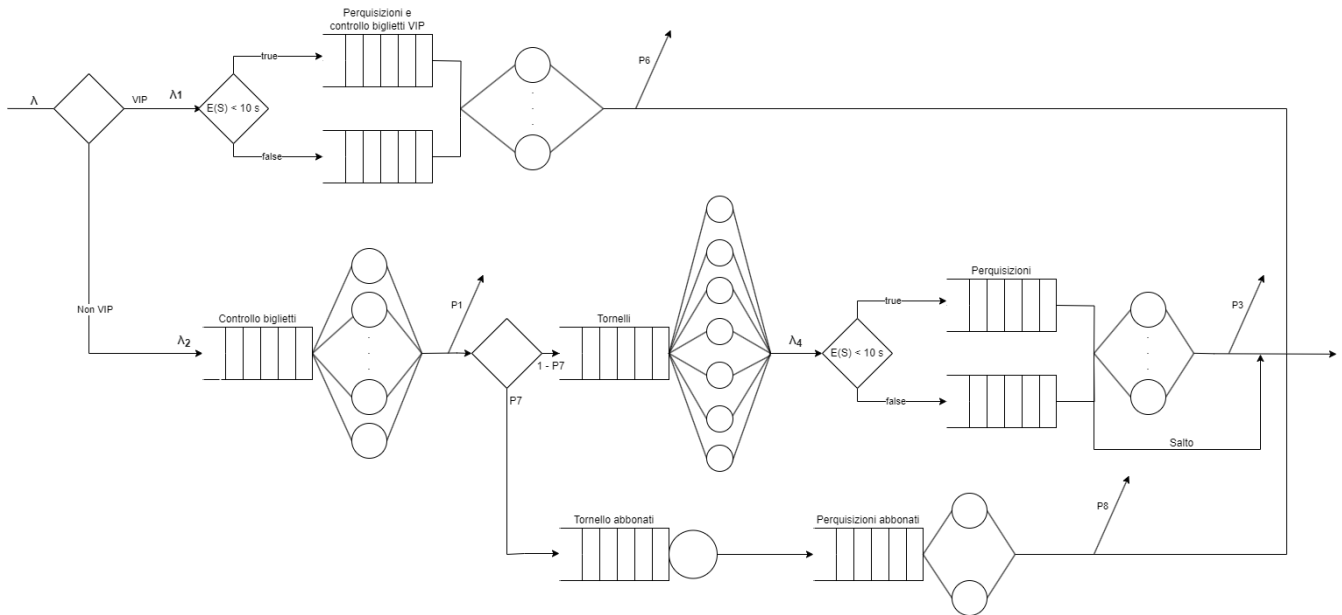
- Utenti VIP
- Utenti standard, che si dividono in:
 - Abbonati
 - Non abbonati

Agli utenti VIP è riservata una coda dedicata, che contiene un unico nodo, *Perquisizioni e controllo biglietti VIP*, realizzato tramite un multi-server (9 serventi) con due classi di priorità size-based, dove alla prima classe appartengono gli utenti che richiedono un tempo di servizio strettamente minore a 10 secondi, mentre alla seconda classe la restante parte degli utenti. Se il controllo del biglietto o la perquisizione non va a buon fine, l'utente viene allontanato: ciò è modellato tramite un abbandono con probabilità fissa $P6 = 11.8\%$ ⁷. Se il controllo del biglietto e la perquisizione vanno a buon fine, l'utente VIP accede allo stadio.

Gli utenti standard accedono inizialmente al nodo *Controllo biglietti*, realizzato tramite una coda $M|M|10$. A seguito del controllo biglietti, può verificarsi un abbandono (con probabilità fissa $P1 = 0.08$), nel caso in cui il controllo biglietti non vada a buon fine. Se, invece, l'utente supera il controllo biglietti, passa al servente successivo. A questo punto, l'utente standard può accedere a due code diverse:

- Se non è abbonato, accede al nodo *Tornelli* ($M|M|7$) e, successivamente, al nodo *Perquisizioni*, realizzato tramite un multi-server (20 serventi) con due code di priorità size-based, analoghe al caso del nodo *Perquisizioni e controllo biglietti VIP*. All'interno di questo nodo, può verificarsi il fenomeno di skip, analogo a quello già implementato nel modello iniziale, nei nodi *Prima perquisizione* e *Seconda perquisizione*. Al termine della perquisizione, c'è un abbandono con probabilità fissa $P3 = 0.02$, che modella le perquisizioni che non vanno a buon fine. Se, invece, la perquisizione è superata, l'utente standard non abbonato accede al settore.
- Se è abbonato, accede ad una coda dedicata. In particolare, accede inizialmente al nodo *Tornello abbonati* ($M|M|1$) e successivamente al nodo *Perquisizione abbonati* ($M|M|2$), a seguito del quale può verificarsi un abbandono con probabilità $P8 = 0.02$. Se la perquisizione va a buon fine, l'utente standard abbonato accede allo stadio.

⁷ $p(\{\text{controllo biglietti fallito}\} \vee \{\text{perquisizione fallita}\}) = p(\{\text{controllo biglietti fallito}\} + p(\{\text{perquisizione fallita}\}) - p(\{\text{controllo biglietti fallito}\} \wedge \{\text{perquisizione fallita}\}) = 0.1 + 0.02 - 0.1 \cdot 0.02$



Stato

Ogni coda può essere vuota o non vuota, mentre ogni servente può essere *busy* o *idle*.

Eventi

Gli eventi che possono verificarsi in ogni servente sono:

- Arrivo di un nuovo utente
- Partenza di un utente servito

Nei nodi *Perquisizione e controllo biglietti VIP*, *Controllo biglietti*, *Perquisizioni* e *Perquisizioni abbonati* può verificarsi anche il seguente evento:

- Abbandono a seguito di controllo non andato a buon fine

Infine, nel nodo *Perquisizioni* può verificarsi anche il seguente evento:

- Salto della perquisizione, in funzione della popolazione corrente in coda

In ogni servente, il job viene eseguito immediatamente se la coda è vuota e se il servente non sta eseguendo job, altrimenti il job viene posto in coda, mantenendo una politica di scheduling FIFO.

9.3 Verifica

Nel processo di verifica è stata utilizzata la prima fascia oraria (i.e., 18:30-19:30) e la configurazione di serveri (*Controllo biglietti e perquisizione VIP*: 9; *Controllo biglietti*: 10; *Tornelli*: 7; *Perquisizioni*: 30; *Tornello abbonati*: 1; *Perquisizioni abbonati*: 2). In particolare, sono stati considerati 30 serveri nel nodo *Perquisizioni* per rimuovere il fenomeno degli *skip* e rendere il sistema studiabile tramite l'analisi.

Analisi del sistema

Controllo biglietti

Sono dati il numero di server $m = 10$, il tempo di servizio medio di ogni server $E(S_i) = 7$ s e il tasso di arrivo $\lambda_{TC} = 0.67011$ utenti/s.

$$\mu_i = \frac{1}{E(S_i)} = 0.05 \text{ utenti/s}$$

$$\rho = \frac{\lambda_{TC}}{m\mu_i} = 0.46908$$

$$E(T_Q) = \frac{P_Q E(S)}{1 - \rho} = \frac{(m\rho)^m}{m!(1 - \rho)} \cdot \frac{1}{\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1 - \rho)}} \cdot \frac{E(S)}{1 - \rho} = 0.03232 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S_i) = 7.03232 \text{ s}$$

$$E(\text{interarrivi}) = \frac{1}{\lambda_{TC}} = 1.49229 \text{ s}$$

$$E(N_S) = \lambda_{TC} E(T_Q) + \rho m = 4.71246 \text{ utenti}$$

Tornelli

Sono dati il numero di server $m = 7$, il tempo di servizio medio di ogni server $E(S_i) = 5$ s e il tasso di arrivo $\lambda_T = \lambda_{TC}(1 - P_1)(1 - P_7) = 0.55485$ utenti/s.

$$\mu_i = \frac{1}{E(S_i)} = 0.2 \text{ utenti/s}$$

$$\rho = \frac{\lambda_T}{m\mu_i} = 0.39632$$

$$E(T_Q) = \frac{P_Q E(S)}{1 - \rho} = \frac{(m\rho)^m}{m!(1 - \rho)} \cdot \frac{1}{\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1 - \rho)}} \cdot \frac{E(S)}{1 - \rho} = 0.03061 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S_i) = 5.03061 \text{ s}$$

$$E(\text{interarrivi}) = \frac{1}{\lambda_T} = 1.80229 \text{ s}$$

$$E(N_S) = \lambda_T E(T_Q) + \rho m = 2.79122 \text{ utenti}$$

Perquisizione

Sono dati il numero di server $m = 30$, il tempo di servizio medio di ogni server $E(S_i) = 20$ s e il tasso di arrivo $\lambda_p = \lambda_T = \lambda_{TC}(1 - P_1)(1 - P_7) = 0.55485$ utenti/s.

$$\mu_i = \frac{1}{E(S_i)} = 0.05 \text{ utenti/s}$$

$$\rho = \frac{\lambda_T}{m\mu_i} = 0.3699$$

$$p_1 = F(10) - F(0) = 1 - e^{-0.5} - 1 + 1 = 0.39347$$

$$p_2 = F(\infty) - F(10) = 0.60653$$

$$E(S_1) = \int_0^{10} \frac{\mu_i t e^{-\mu_i t} dt}{p_1} = 4.58506 \text{ s}$$

$$E(S_2) = \int_{10}^{+\infty} \frac{\mu_i t e^{-\mu_i t} dt}{p_1} = 30 \text{ s}$$

$$\rho_1 = \frac{\lambda_p p_1 E(S_1)}{30} = 0.03337$$

$$\rho_2 = \frac{\lambda_p p_2 E(S_2)}{30} = 0.33653$$

$$E(T_Q) = \frac{p_1 P_{Q1} E(S)}{1 - \rho_1} + \frac{p_2 P_{Q2} E(S)}{(1 - \rho)(1 - \rho_1)} = 2.07189 \cdot 10^{-7} \text{ s}$$

$$E(T_S) = E(T_Q) + E(S_i) \approx 5 \text{ s}$$

$$E(\text{interarrivi}) = \frac{1}{\lambda_p} = 1.80229 \text{ s}$$

$$E(N_S) = \lambda_p E(T_Q) + \rho m = 11.097 \text{ utenti}$$

Tornello abbonati

Sono dati il numero di server $m = 1$, il tempo di servizio medio del server $E(S) = 5$ s e il tasso di arrivo $\lambda_{ST} = \lambda_{TC}(1 - P_1)P_7 = 0.06165$ utenti/s.

$$\mu = 0.2 \text{ s}$$

$$\rho = \frac{\lambda_{ST}}{\mu} = 0.30825$$

$$E(T_Q) = \frac{\rho E(S)}{1 - \rho} = 2.22805 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S) = 7.22805 \text{ s}$$

$$E(N_Q) = \lambda_{ST} E(T_Q) = 0.13736 \text{ utenti}$$

$$E(N_S) = E(N_Q) + \rho = 0.44561 \text{ utenti}$$

$$E(\text{interarrivi}) = \frac{1}{\lambda_{ST}} = 16.2206 \text{ s}$$

Perquisizioni abbonati

Sono dati il numero di server $m = 2$, il tempo di servizio medio di ogni server $E(S_i) = 20$ s e il tasso di arrivo $\lambda_{SP} = \lambda_{ST} = 0.06165$ utenti/s.

$$\mu_i = \frac{1}{E(S_i)} = 0.05 \text{ utenti/s}$$

$$\rho = \frac{\lambda_{SP}}{m\mu_i} = 0.6165$$

$$E(T_Q) = \frac{P_Q E(S)}{1 - \rho} = \frac{(m\rho)^m}{m!(1 - \rho)} \cdot \frac{1}{\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1 - \rho)}} \cdot \frac{E(S)}{1 - \rho} = 12.26182 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S_i) = 32.26182 \text{ s}$$

$$E(\text{interarrivi}) = \frac{1}{\lambda_{SP}} = 16.2206 \text{ s}$$

$$E(N_S) = \lambda_{SP} E(T_Q) + \rho m = 1.98894 \text{ utenti}$$

Controllo biglietti e perquisizione VIP

Sono dati il numero di server $m = 9$, il tempo di servizio medio di ogni server $E(S_i) = 27$ s e il tasso di arrivo $\lambda_V = 0.05122$ utenti/s.

$$\mu_i = \frac{1}{E(S_i)} = 0.03704 \text{ utenti/s}$$

$$\rho = \frac{\lambda_V}{m\mu_i} = 0.15365$$

$$p_1 = F(10) - F(0) = 1 - e^{-0.3704} - 1 + 1 = 0.30954$$

$$p_2 = F(\infty) - F(10) = 0.69046$$

$$E(S_1) = \int_0^{10} \frac{\mu_i t e^{-\mu_i t} dt}{p_1} = 4.69207 \text{ s}$$

$$E(S_2) = \int_{10}^{+\infty} \frac{\mu_i t e^{-\mu_i t} dt}{p_1} = 36.99774 \text{ s}$$

$$\rho_1 = \frac{\lambda_V p_1 E(S_1)}{30} = 0.00827$$

$$\rho_2 = \frac{\lambda_V p_2 E(S_2)}{30} = 0.14538$$

$$E(T_Q) = \frac{p_1 P_{Q1} E(S)}{1 - \rho_1} + \frac{p_2 P_{Q2} E(S)}{(1 - \rho)(1 - \rho_1)} = 2.7611 \cdot 10^{-5} \text{ s}$$

$$E(T_S) = E(T_Q) + E(S_i) \approx 27 \text{ s}$$

$$E(\text{interarrivi}) = \frac{1}{\lambda_V} = 19.5229 \text{ s}$$

$$E(N_S) = \lambda_V E(T_Q) + \rho m = 1.38285 \text{ utenti}$$

Per la fase di verifica, sono state prodotte due simulazioni a orizzonte infinito tramite *Batch Means*:

- Per la coda standard: $b = 7168$, $k = 128$.
- Per la coda VIP: $b = 128$, $k = 30$.

La scelta della dimensione del batch, cioè del valore del parametro b , è stata guidata dal criterio di *Banks et al. [2001]*⁸, secondo cui il valore di b deve aumentare fintanto che il valore di lag-autocorrelation, tra le medie dei batch, non scende sotto 0.2. Ciò è stato verificato tramite l'analisi dei dati di output implementata nella classe *Acs.java*.

Le seguenti tabelle confrontano i valori analitici con i valori dati dalla simulazione, per ogni nodo del sistema.

Statistica	Risultato analitico	Risultato simulazione
$E(T_S)$	7.03232 s	7.02447 +/- 0.01493 s
$E(T_Q)$	0.03232 s	0.03067 +/- 0.00222 s
ρ	0.46908	0.46935 +/- 0.00131
tempo di interarrivo	1.49299 s	1.49030 +/- 0.00294 s
$E(S)$	7 s	6.99379 +/- 0.01410 s
$E(N_S)$	4.71246	4.71408 +/- 0.01374

Tabella 15: Controllo biglietti

Statistica	Risultato analitico	Risultato simulazione
$E(T_S)$	5.03061 s	5.01854 +/- 0.01177 s
$E(T_Q)$	0.03061 s	0.03009 +/- 0.00170 s
ρ	0.39632	0.39518 +/- 0.00139
tempo di interarrivo	1.80229 s	1.80372 +/- 0.00459 s
$E(S)$	5 s	4.98844 +/- 0.01118 s
$E(N_S)$	2.79122	2.78297 +/- 0.01015

Tabella 16: Tornelli

Statistica	Risultato analitico	Risultato simulazione
$E(T_S)$	20 s	20.02408 +/- 0.04243 s
$E(T_Q)$	0 s	0.00000 +/- 0.00000 s
ρ	0.3699	0.37014 +/- 0.00131
tempo di interarrivo	1.80229 s	1.80376 +/- 0.00456 s
$E(S)$	20 s	20.02418 +/- 0.04207 s
$E(N_S)$	11.097	11.10387 +/- 0.03885

Tabella 17: Perquisizione

⁸ Banks, Carson, Nelson, and Nicol (2001, page 438) recommend that the batch size be increased until the lag one autocorrelation between batch means is less than 0.2.

Statistica	Risultato analitico	Risultato simulazione
$E(T_S)$	7.22805 s	7.25774 +/- 0.09028 s
$E(T_Q)$	2.22805 s	2.25007 +/- 0.07049 s
ρ	0.30825	0.30976 +/- 0.00267
tempo di interarrivo	16.2206 s	16.18520 +/- 0.09847 s
$E(S)$	5 s	5.00759 +/- 0.03143 s
$E(N_S)$	0.44561	0.44924 +/- 0.00685

Tabella 18: Tornello abbonati

Statistica	Risultato analitico	Risultato simulazione
$E(T_S)$	32.26182 s	31.88732 +/- 0.61449 s
$E(T_Q)$	12.26182 s	12.00487 +/- 0.54003 s
ρ	0.6165	0.61491 +/- 0.00541
tempo di interarrivo	16.2206 s	16.18547 +/- 0.09883 s
$E(S)$	20 s	19.88220 +/- 0.13168 s
$E(N_S)$	1.98894	1.97483 +/- 0.04348

Tabella 19: Perquisizioni abbonati

Statistica	Risultato analitico	Risultato simulazione
$E(T_S)$	27 s	26.89704 +/- 0.44126 s
$E(T_Q)$	0 s	0.00000 +/- 0.00000 s
ρ	0.15365	0.15424 +/- 0.00318
tempo di interarrivo	19.5229 s	19.48965 +/- 0.27945 s
$E(S)$	27 s	26.89571 +/- 0.44544 s
$E(N_S)$	1.38285	1.38815 +/- 0.02837

Tabella 20: Controllo biglietti e perquisizioni VIP

9.4 Validazione

Pur avendo contattato l'azienda, non è stato possibile reperire dati storici riguardanti il sistema di interesse.

Di conseguenza, al fine di validare il sistema, sono stati riproposti i seguenti controlli di consistenza, già effettuati nel modello standard:

- Al crescere del tasso di arrivo (e.g. passando dalla fascia oraria 18:30 – 19:30 alla più congestionata fascia oraria 19:30 – 20:30), si ha un aumento dei tempi di servizio, in ogni servente.
- Al crescere del tasso di arrivo, si ha un aumento nei salti alle perquisizioni.

Per realizzare i controlli di consistenza, sono state utilizzate le simulazioni a orizzonte infinito introdotte nella verifica, ottenendo i seguenti risultati:

- Per il primo controllo di consistenza, sono stati confrontati i tempi di risposta medi ottenuti su tutti i serventi tramite *Batch Means*, considerando una volta la fascia oraria 18:30-19:30 e una volta la fascia oraria 19:30-20:30.

La configurazione di server è la seguente: (*Controllo biglietti e perquisizione VIP: 9; Controllo biglietti: 10; Tornelli: 7; Perquisizioni: 20; Tornello abbonati: 1; Perquisizioni abbonati: 2*).

Nodo del sistema	$E(T_S)_{18:30-19:30}$	$E(T_S)_{19:30-20:30}$
Controllo biglietti	7.03108 s	206904.37027 s
Tornelli	5.01853 s	7.62156 s
Perquisizione	20.05048 s	36332.77442 s
Tornello abbonati	7.26365 s	15.20231 s
Perquisizioni abbonati	31.78209 s	97883.97138 s
Controllo biglietti e perquisizioni VIP	26.89704 s	26.98388 s

Tabella 21: Confronto tra i tempi di risposta medi delle prime due fasce orarie

Dai risultati, si vede come il passaggio dalla prima alla seconda fascia oraria, a parità di configurazione, porti ad un'esplosione dei tempi di servizio medi per quanto riguarda i serventi *Controllo biglietti*, *Perquisizioni* e *Perquisizioni abbonati*, mentre si ha una crescita più contenuta nei tempi di risposta degli altri nodi. In particolare, si vede come nel nodo *Controllo biglietti e perquisizioni VIP* il tempo di risposta cresce di una quantità trascurabile: ciò dimostra il fatto che la quantità di serventi impiegati in questo nodo è sovradimensionata, cioè si possono mantenere ottime prestazioni diminuendo il numero di serventi.

- Per verificare il secondo controllo di consistenza, è stata impiegata la seguente configurazione: (*Controllo biglietti e perquisizione VIP: 9; Controllo biglietti: 30; Tornelli: 7; Perquisizioni: 20; Tornello abbonati: 1; Perquisizioni abbonati: 2*). In particolare, è stato aumentato il numero di serventi del nodo *Controllo biglietti*, per evitare che questo agisse da *bottleneck*. Di seguito sono mostrati i risultati ottenuti:

Nodo del sistema	Numero di skip (18:30-19:30)	Numero di skip (19:30-20:30)
Perquisizione	0.00000 +/- 0.00000	1418.57812 +/- 21.11213

Tabella 22: Confronto tra la quantità di salti nelle prime due fasce orarie

9.5 Progettazione degli esperimenti

I requisiti di QoS che il modello migliorativo deve rispettare sono i seguenti:

- Tempo di accesso allo stadio per gli utenti standard abbonati inferiore a 15 minuti.
- Tempo di accesso allo stadio per gli utenti standard non abbonati inferiore a 25 minuti.
- Tempo di accesso allo stadio per gli utenti VIP inferiore a 3 minuti.
- Percentuale di utenti perquisita pari al 100%.

Al fine di trovare la configurazione che rispetti tutti i requisiti di cui sopra, sono state effettuate delle simulazioni a orizzonte finito.

9.6 Simulazioni e analisi dei risultati

Simulazione a orizzonte finito

Primo requisito di QoS

Inizialmente, è stata effettuata una simulazione a orizzonte infinito utilizzando la configurazione di partenza, cioè la seguente: (Controllo biglietti e perquisizione VIP: 9; Controllo biglietti: 10; Tornelli: 7; Perquisizioni: 20; Tornello abbonati: 1; Perquisizioni abbonati: 2), ottenendo i seguenti tempi di risposta:

Nodo del sistema	$E(T_S)$
Controllo biglietti	908.42532 +/- 6.72211 s
Tornello abbonati	12.75605 +/- 0.27118 s
Perquisizioni abbonati	862.05767 +/- 32.46928 s

Tabella 23: Risultati da 150 repliche con la configurazione (9, 10, 7, 20, 1, 2)

Il tempo medio di risposta totale è pari a 1783.23904 secondi, cioè circa 30 minuti. Di conseguenza, il primo requisito non è soddisfatto.

Dai risultati, risulta evidente che il sistema è congestionato nei nodi *Controllo biglietti* e *Perquisizione abbonati*. Passando alla configurazione (Controllo biglietti e perquisizione VIP: 9; Controllo biglietti: 11; Tornelli: 7; Perquisizioni: 20; Tornello abbonati: 1; Perquisizioni abbonati: 3), si ottengono i seguenti risultati:

Nodo del sistema	$E(T_S)$
Controllo biglietti	679.29119 +/- 5.85067 s
Tornello abbonati	15.07484 +/- 0.37986 s
Perquisizioni abbonati	88.98036 +/- 9.04597 s

Tabella 24: Risultati da 150 repliche con la configurazione (9, 11, 7, 20, 1, 3)

In questo caso, il tempo medio di risposta totale è di 783.34639 secondi, pari a circa 13 minuti.

Di conseguenza, la configurazione (Controllo biglietti e perquisizione VIP: 9; Controllo biglietti: 11; Tornelli: 7; Perquisizioni: 20; Tornello abbonati: 1; Perquisizioni abbonati: 3) rispetta il primo requisito di QoS.

Secondo requisito di QoS

In relazione ai server relativi agli utenti standard non abbonati, la configurazione (*Controllo biglietti e perquisizione VIP: 9; Controllo biglietti: 10; Tornelli: 7; Perquisizioni: 20; Tornello abbonati: 1; Perquisizioni abbonati: 2*) produce i seguenti risultati:

Nodo del sistema	$E(T_s)$
Controllo biglietti	908.42532 +/- 6.72211 s
Tornelli	7.08473 +/- 0.06882 s
Perquisizioni	342.85646 +/- 6.08022 s

Tabella 25: Risultati da 150 repliche con la configurazione (9, 10, 7, 20, 1, 2)

Il tempo medio di risposta totale è 1258.39651 secondi, cioè circa 21 minuti. Di conseguenza, il secondo requisito è già rispettato dalla configurazione standard.

Terzo requisito di QoS

Effettuando una simulazione a orizzonte finito con 150 repliche, si vede che il tempo di risposta medio dell'unico nodo presente nella coda VIP è pari a 27.07618 secondi.

Questo tempo rispetta ampiamente il requisito di 3 minuti, di conseguenza, al fine di risparmiare risorse continuando però a rispettare il requisito, il numero di server del nodo *Controllo biglietti e perquisizioni VIP* è stato ridotto da 9 a 4, ottenendo un tempo di risposta medio di 73.50168 secondi, inferiore al requisito di 3 minuti.

Passando a 3 server, si ottiene un tempo di risposta medio di 351.80985 secondi, che non rispetta il requisito di QoS.

Di conseguenza, la configurazione ottimale prevede 4 server, nel nodo *Controllo biglietti e perquisizioni VIP*.

Quarto requisito di QoS

Utilizzando la configurazione (*Controllo biglietti e perquisizione VIP: 4; Controllo biglietti: 11; Tornelli: 7; Perquisizioni: 20; Tornello abbonati: 1; Perquisizioni abbonati: 3*), dalla simulazione a orizzonte infinito otteniamo che il numero medio di skip, nel nodo *Perquisizioni*, è pari a 1439,13333. Di conseguenza, non tutti gli utenti sono perquisiti.

Aggiungendo tre server al nodo perquisizioni, si ottengono, in media, 7,96667 salti.

La configurazione che raggiunge l'obiettivo, quindi, prevede l'aggiunta di quattro server al nodo *Perquisizione*.

Conclusioni

La configurazione che consente di raggiungere tutti i requisiti di QoS è la seguente:

- (*Controllo biglietti e perquisizione VIP: 4; Controllo biglietti: 11; Tornelli: 7; Perquisizioni: 24; Tornello abbonati: 1; Perquisizioni abbonati: 3*)

Nella seguente tabella sono riassunti il tempo di risposta totale di ogni servente e il numero di skip, ottenuti utilizzando questa configurazione:

Nodo del sistema	$E(T_s)$	Numero di skip
Controllo biglietti	679,29119 +/- 5,85067 s	/
Tornelli	11,01014 +/- 0,31485 s	/
Perquisizioni	148,93589 +/- 4,54515 s	0,00000 +/- 0,00000
Tornello abbonati	15,01485 +/- 0,40832	/
Perquisizioni abbonati	88,83784 +/- 8,60704	/
Controllo biglietti e perquisizioni VIP	73,50168 +/- 5,10871	/

Tabella 26: Risultati da 150 replicazioni con la configurazione (4, 11, 7, 24, 1, 3)

- Gli utenti standard non abbonati, in media, accedono allo stadio in 839.23722 secondi, cioè circa 14 minuti.
- Gli utenti standard abbonati, in media, accedono allo stadio in 783.14388 secondi, cioè circa 13 minuti.
- Gli utenti VIP, in media, accedono allo stadio in 73.50168 secondi, cioè in poco più di un minuto.
- Tutti gli utenti sono perquisiti.

Si noti che il quarto requisito, cioè che tutti gli utenti siano perquisiti, conduce ad una diminuzione del tempo di risposta del nodo *Perquisizione*, in quanto devono essere aggiunti serventi in questo nodo. Ciò porta ad una decrescita del tempo di accesso allo stadio per gli utenti standard non abbonati, che si avvicina molto a quello per gli utenti abbonati, di fatto limitando il vantaggio di questi ultimi per quanto riguarda i minori tempi di ingresso.

10. Conclusioni

Analisi dei costi e dei benefici

La configurazione di partenza del modello iniziale è:

- (*Controllo biglietti: 10; Prima perquisizione: 20; Tornelli: 8; Seconda perquisizione: 20; Controllo biglietti VIP: 4; Perquisizione VIP: 5*)

La configurazione finale del modello migliorativo, invece, è:

- (*Controllo biglietti e perquisizione VIP: 4; Controllo biglietti: 11; Tornelli: 7; Perquisizioni: 24; Tornello abbonati: 1; Perquisizioni abbonati: 3*)

In totale, si passa da 67 serventi totali a 50. In particolare, per quanto riguarda gli steward, si passa dai 14 presenti nel modello iniziale agli 11 presenti nel modello migliorativo⁹, mentre per quanto riguarda le forze dell'ordine, si passa da 45 agenti nel modello iniziale a 31 nel modello migliorativo. Il numero di tornelli resta invariato.

In termini di costi, considerando un compenso medio di 55 €/evento per gli steward e 80 €/evento per i poliziotti, si passa da 4370 €/evento a 3085 €/evento, ottenendo quindi un risparmio di 1285 €/evento.

Per quanto concerne i tempi di accesso allo stadio, la configurazione del modello iniziale prevede:

- Tempo di accesso allo stadio per utenti standard: 1835.39442 secondi, cioè circa 31 minuti.
- Tempo di accesso allo stadio per utenti VIP: 29.46179 secondi.

Invece, il modello migliorativo prevede:

- Tempo di accesso allo stadio per utenti standard non abbonati: 839.23722 secondi, cioè circa 14 minuti.
- Tempo di accesso allo stadio per utenti standard abbonati: 783.14388 secondi, cioè circa 13 minuti.
- Tempo di accesso allo stadio per utenti VIP: 73.50168 secondi, cioè circa un minuto.

Di conseguenza, i tempi medi di accesso allo stadio, per quanto riguarda gli utenti standard, sono stati più che dimezzati. Per quanto riguarda invece gli utenti VIP, i tempi di ingresso al settore sono aumentati del 150%; ciò dipende dalla diminuzione dei serventi associati al nodo *Controllo biglietti e perquisizione VIP* rispetto a quanti erano assegnati ai due nodi *Controllo biglietti VIP* e *Perquisizioni VIP*. Infatti, mantenendo invariato il numero di serventi, il tempo medio di ingresso allo stadio sarebbe di 27.07618 secondi, come si vede nella sezione precedente.

In conclusione, la proposta di modello migliorativo porta ad una riduzione dei costi di circa il 30% e ad una riduzione dei tempi di risposta totali, rispettivamente per utenti standard non abbonati e abbonati, di circa il 55% e il 58%, mantenendo prestazioni lievemente peggiori ma ancora accettabili per quanto riguarda gli utenti VIP.

⁹ Si presuppone che il personale assegnato al controllo biglietti e perquisizione VIP sia interamente composto da agenti di polizia.