Performance Analysis of a Web App Workflow

Daniele La Prova, Elisa Verza

Universita' degli Studi di Roma Tor Vergata

Roadmap

Context

- In questo lavoro presentiamo un'analisi delle prestazioni di una applicazione di e-commerce sviluppata usando uno stack tecnologico WEB.
- Il lavoro è ispirato a uno studio analogo delle perfomance effettuato da Serazzi (2024).

Context

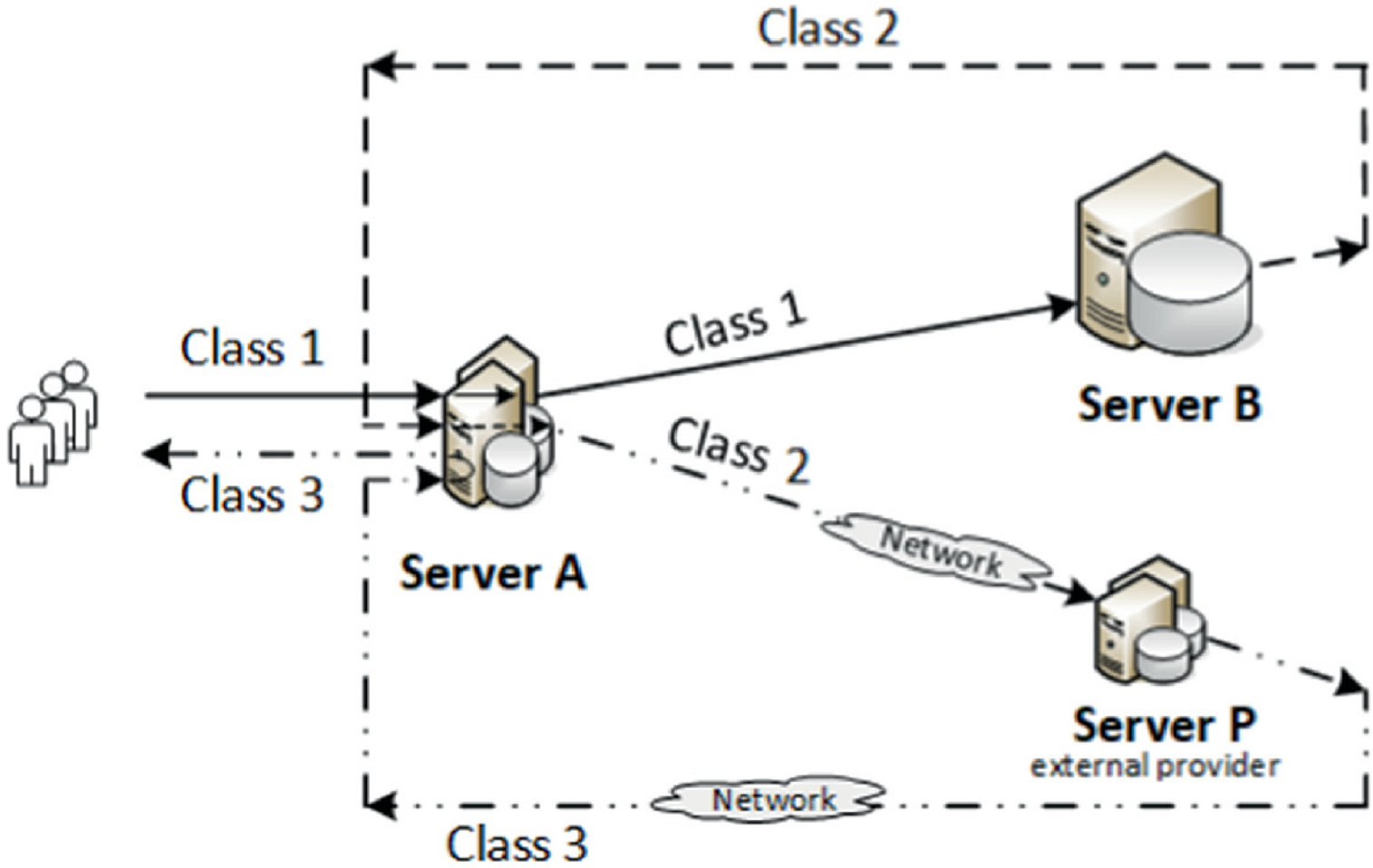


Figure 1: Vista Bird-Eye dell'architettura distribuita della Web App. (Serazzi, 2024)

Roadmap

Objectives(1/2)

1. **Obiettivo 1**: implementare un modello in grado di rappresentare il sistema oggetto di studio e misurare:
   - tempo di risposta,
   - popolazione media,
   - throughput medio.

   Le metriche devono essere considerate sia localmente (per singolo nodo) che globalmente (su tutto il sistema).
2. **Obiettivo 2**: valutare l'impatto di un sistema di autenticazione a due fattori per rendere i pagamenti più sicuri, considerando le stesse metriche dell'obiettivo precedente.

Objectives(2/2)

3. **Obiettivo 3**: osservare il comportamento del sistema in caso di carico di richieste maggiore:
   - Incremento da 4320 job/h (1.2 job/s) a circa 5000 job/h (1.4 job/s).

   Il confronto viene eseguito sulle stesse metriche dei primi due obiettivi.
4. **Obiettivo 4**: migliorare il sistema individuando eventuali bottleneck per aumentarne le prestazioni.

Conceptual Model: Schema del sistema

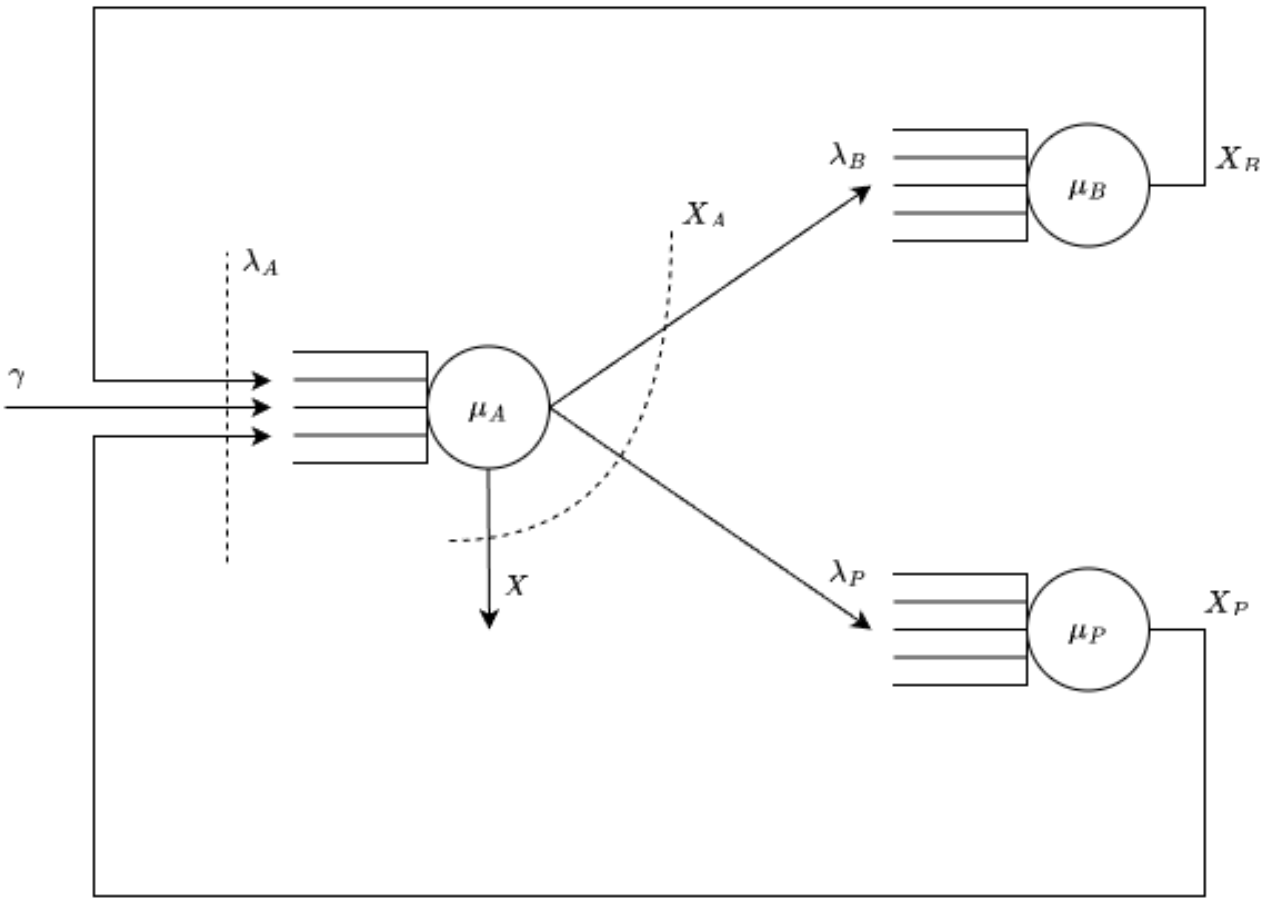


Figure 2: Diagramma a reti di code del sistema.

Conceptual Model: Job e classi

Ai job è stata assegnata una classe seguendo il seguente criterio:

- **Classe 1**: job in arrivo dall'esterno, restano in questa classe fino all'ingresso in servizio in B.
- **Classe 2**: job in arrivo al server A tramite feedback del server B, restano in questa classe fino all'ingresso in servizio in P.
- **Classe 3**: job in arrivo al server A tramite feedback del server P. Questa classe include i job che escono dal sistema.

Conceptual Model: Variabili di stato

Le variabili di stato che meglio descrivono il sistema sono:

- $N_{\text{node, class}}$: numero di job di classe *class* $\in \{1, 2, 3\}$ nel nodo *node* $\in \{A, B, P\}$.
- C_j con $j \in [0, \infty)$: valore che indica la classe di appartenenza del job j.

La simulazione utilizzata è una *next-event simulation*. Gli eventi sono:

- **Arrival**: caratterizzati da server di destinazione e classe del job in arrivo.
- **Departure**: caratterizzati da server di partenza e classe del job in partenza.

Conceptual Model: Evoluzione degli eventi

Gli eventi nei server A, B, e P seguono un pattern generale:

- **Arrival:**
  - Incremento della variabile di stato del nodo relativo, $N_{\text{node,class}}$.
  - Generazione dell'evento *Departure* per il job appena arrivato.
  - Se l'arrivo è dall'esterno C_j viene posta ad 1.
- **Departure:**
  - Decremento della variabile di stato $N_{\text{node,class}}$.
  - Incremento della variabile C_j.
  - Generazione di un evento *Arrival* nel nodo successivo o uscita dal sistema (se il job è di classe 3 in A).

Specification Model: Variabili indipendenti (Serazzi, 2024)

- Distribuzioni di probabilità dei servizi: Esponenziali
- Distribuzioni di probabilità degli arrivi esterni: Esponenziali
- Rate medi di arrivi esterni: valori che spaziano da 0.50 a 1.20 con un passo di 0.05 job/s, per poi estendere fino a 1.40 job/s nel caso di carico pesante
- Politiche di scheduling: PS

Stations	Classes		
	1	2	3
Server A (Login, Front end, ...)	0.2	0.4	0.1
Server B (Web App Serv., DBs, ...)	0.8	0	0
Server P (Payment Provider)	0	0.4	0

Stations	Classes		
	1	2	3
Server A (Login, Front end, ...)	0.2	0.4	0.15
Server B (Web App Serv., DBs, ...)	0.8	0	0
Server P (Payment Provider)	0	0.7	0

(a) Tempi di servizio medi per classe di job in ogni nodo, nella versione vanilla (sinistra) e con 2FA (destra).

(b) Routing Matrix per jobs di una certa classe che arrivano in un server.

	class 1	class 2	class3
Server A	**Server B**	**Server P**	**EXIT**
Server B	**Server A**		
Server P		**Server A**	

Specification Model: Variabili dipendenti

- Tempo di risposta T: Calcolato come differenza temporale tra istante di entrata e di uscita di una richiesta in un nodo
- Popolazione N: Numero richieste all'interno di un nodo
- Throughput X: Numero di richieste soddisfatte sull'unità di tempo dal nodo
- Utilizzazione ρ: Proporzione di tempo in cui il nodo è occupato a smaltire richieste durante un periodo di osservazione

Computational Model: Scheduler Next-Event

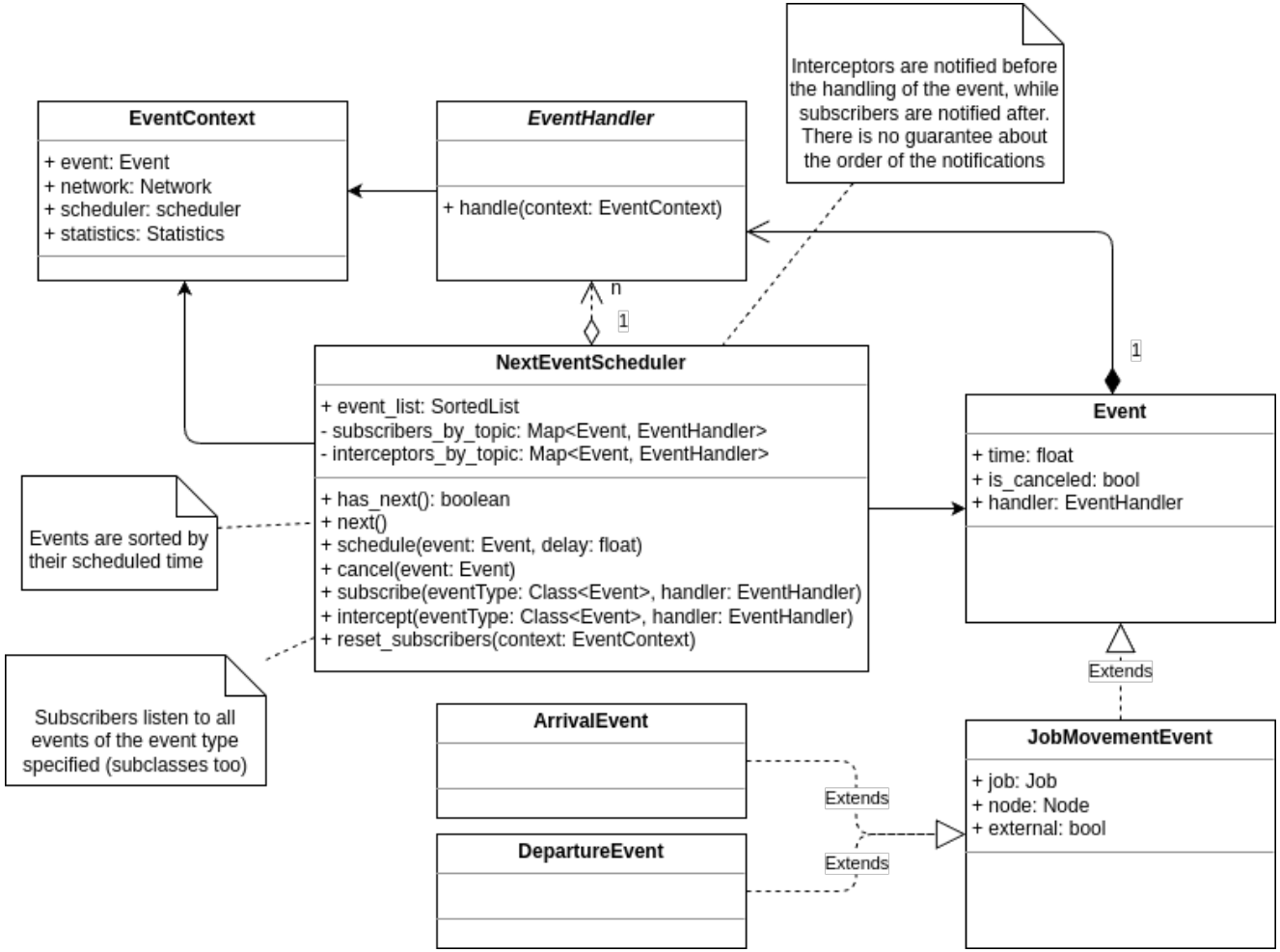


Figure 4: Class diagram dello scheduler Next-Event.

Computational Model: Event-Oriented Programming

- medie calcolate secondo algoritmo di Welford (Lawrence M. Leemis, 2004)
- Stimatori:
  - `ArrivalsGeneratorSubscriber`
  - `BusytimeEstimator`
  - `ObservationTimeEstimator`
  - `CompletionsEstimator`
  - `ResponseTimeEstimator`
  - `PopulationEstimator`
- altri listeners:
  - `ArrivalsGeneratorSubscriber`
  - `BatchMeansInterceptor`

Computational Model: Job Movements

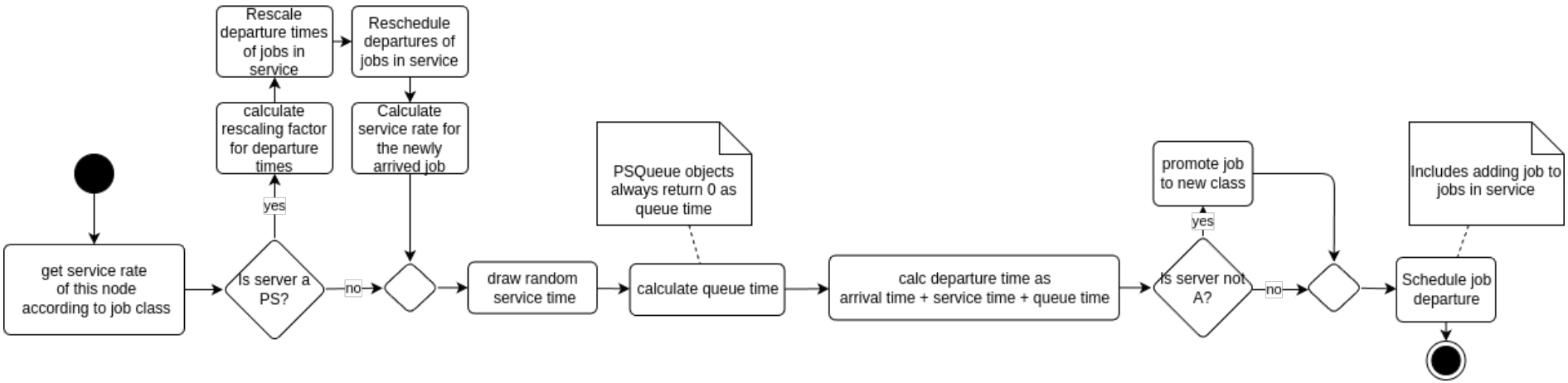


Figure 5: Activity diagram dell'handler degli arrivi.

Theorem (Rescaled Remaining Service Time)

$$S^*_{rem} = S_{rem} \frac{N^*}{N} \tag{1}$$

Computational Model: Job Movements

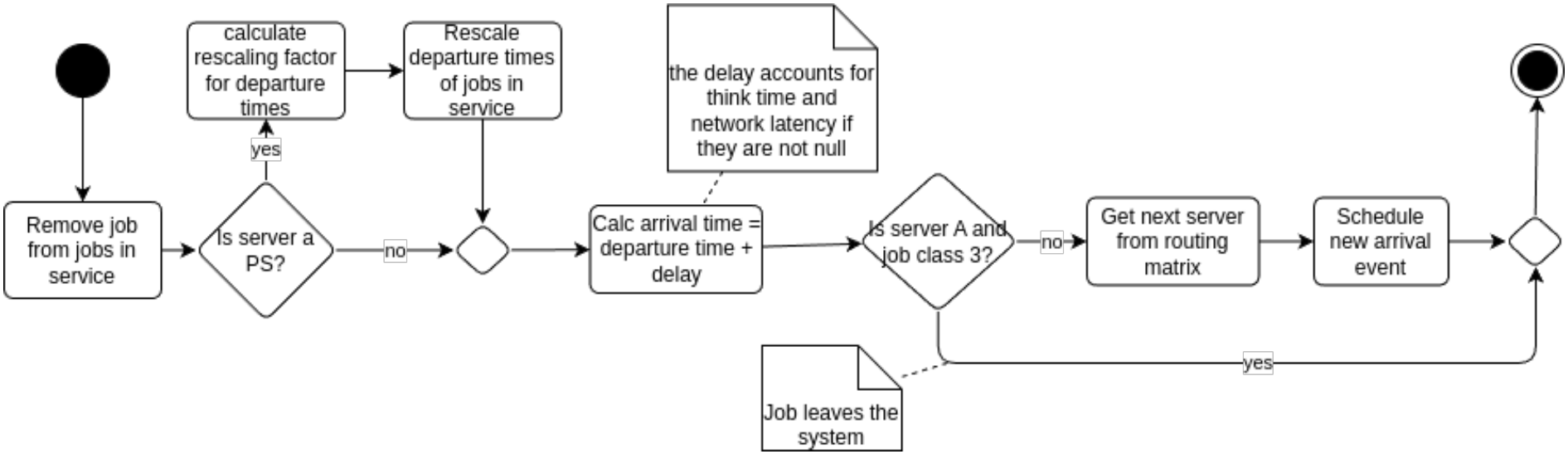


Figure 5: Activity diagram dell'handler delle partenze.

Theorem (Rescaled Remaining Service Time)

$$S^*_{rem} = S_{rem} \frac{N^*}{N} \tag{1}$$

Computational Model: Stochastic processes

- La simulazione dei processi casuali è stata implementata usando il modulo `rngs` (Lawrence M. Leemis, 2004)
- Per ogni processo casuale indipendente simulato nel sistema è selezionato uno stream dedicato
- I processi pseudo-casuali indipendenti implementati nel sistema sono:
  - Gli arrivi dall'esterno del sistema al server A;
  - I servizi del server A;
  - I servizi del server B;
  - I servizi del server P;

Computational Model: Simulation Runs

- Una simulazione è implementata come un oggetto `Simulation`:
  - `BatchMeansSimulation`: Iscrive alla simulazione decorata un `BatchMeansInterceptor` per implementare una simulazione Batch Means;
  - `ReplicatedSimulation`: Esegue una serie di repliche di simulazione in sequenza, impostando il seed iniziale della successiva come l'ultimo valore della sequenza prng della precedente.

Verify: Assunzioni modello analitico

- Il modello analitico considera il server A con un tasso di servizio medio unico per tutte le classi di job:

$$\mu_A = \sum_{c=1}^{3} \mu_{A,c} \cdot p_{A,c}$$

  - $\mu_{A,c}$: tasso di servizio di A per la classe c.
  - $p_{A,c}$: probabilità di trovare un job di classe c in A ($\frac{1}{3}$ per ciascuna classe).

Verify: Assunzioni modello analitico

- Il modello analitico considera il server A con un tasso di servizio medio unico per tutte le classi di job:

$$\mu_A = \sum_{c=1}^{3} \mu_{A,c} \cdot p_{A,c}$$

  - $\mu_{A,c}$: tasso di servizio di A per la classe c.
  - $p_{A,c}$: probabilità di trovare un job di classe c in A ($\frac{1}{3}$ per ciascuna classe).

- Assunto che il flusso dei job è bilanciato, vengono calcolati i rate di arrivo ai server (λ_s) risolvendo il bilanciamento dei flussi:

$$\lambda_A = \gamma + X_B + X_P, \quad \lambda_B = X_B, \quad \lambda_P = X_P$$

Verify: Indici locali modello analitico

- **Stabilità del sistema**: verificata con l'utilizzazione (ρ_s):

$$\rho_s = \frac{\lambda_s}{\mu_s}, \quad \text{per ogni server.}$$

Verify: Indici locali modello analitico

- **Stabilità del sistema**: verificata con l'utilizzazione (ρ_s):

$$\rho_s = \frac{\lambda_s}{\mu_s}, \quad \text{per ogni server.}$$

- **Indici locali**:
  - Tempo di risposta medio per server:

$$E[T]_s = \frac{1}{(1-\rho_s)\cdot\mu_s}$$

  - Popolazione media per server (legge di Little):

$$E[N]_s = E[T]_s \cdot \lambda_s$$

Verify: Indici globali modello analitico

- **Visite medie per server**:

$$v_s = \frac{\lambda_s}{\gamma}$$

Verify: Indici globali modello analitico

- **Visite medie per server**:

$$v_s = \frac{\lambda_s}{\gamma}$$

- **Indici globali**:
  - Tempo di risposta medio:

$$E[T] = \sum_{s=A}^{P} E[T]_s \cdot v_s$$

  - Popolazione media (legge di Little):

$$E[N] = \gamma \cdot E[T]$$

  - Throughput (server A, job classe 3):

$$X = X_A \cdot p_{A,3}$$

Validation

- In assenza di dati sperimentali riguardanti una implementazione reale della Web App non è possibile effettuare una validazione vera e propria.
- abbiamo validato i nostri risultati con quelli di Serazzi (2024)

Simulation Studies: Tipologie di simulazione

- **Tipologia delle simulazioni**:
  - **Simulazioni ad orizzonte infinito**: Batch-Means con:
    - 100 batch
    - 64 job per batch
  - **Simulazioni transitorie**:
    - 5 run al variare del seed
    - 64 job per run

Simulation Studies: Parametri di simulazione (1/2)

1. **Obiettivo 1**: Valutare il comportamento dei server nella configurazione base

Server	Classe 1	Classe 2	Classe 3
A	5	2.5	10
B	1.25	0	0
C	0	2.5	0

2. **Obiettivo 2**: Analizzare l'impatto dell'**autenticazione a due fattori**
   - Server A (classe 3): $6.6667\ job/s$
   - Server P: $1.4285\ job/s$
   - Server B: invariato.

Simulation Studies: Parametri di simulazione (2/2)

3. **Obiettivo 3**: Analizzare l'impatto di un nuovo **carico pesante**:
   - Configurazione base per i tassi di servizio.
4. **Obiettivo 4**: Miglioramento delle performance
   - Variazione del tasso di servizio del **server B**.
   - Configurazione base per gli altri server.

Simulation Studies: Grafici

Abbiamo analizzato tramite grafici:

- **Gli intervalli di confidenza (95%)**:

$$CI = \bar{x} \pm z \cdot \frac{\sigma}{\sqrt{n}}$$

  - $\bar{x}$: media campionaria.
  - z: valore critico della distribuzione normale standard ($z = 1.96$).
  - σ: deviazione standard.
  - n: numero di sample points.

- **Distribuzione delle metriche**:
  - Analisi della distribuzione per run.

Roadmap

Objective 1: Infinite Horizon

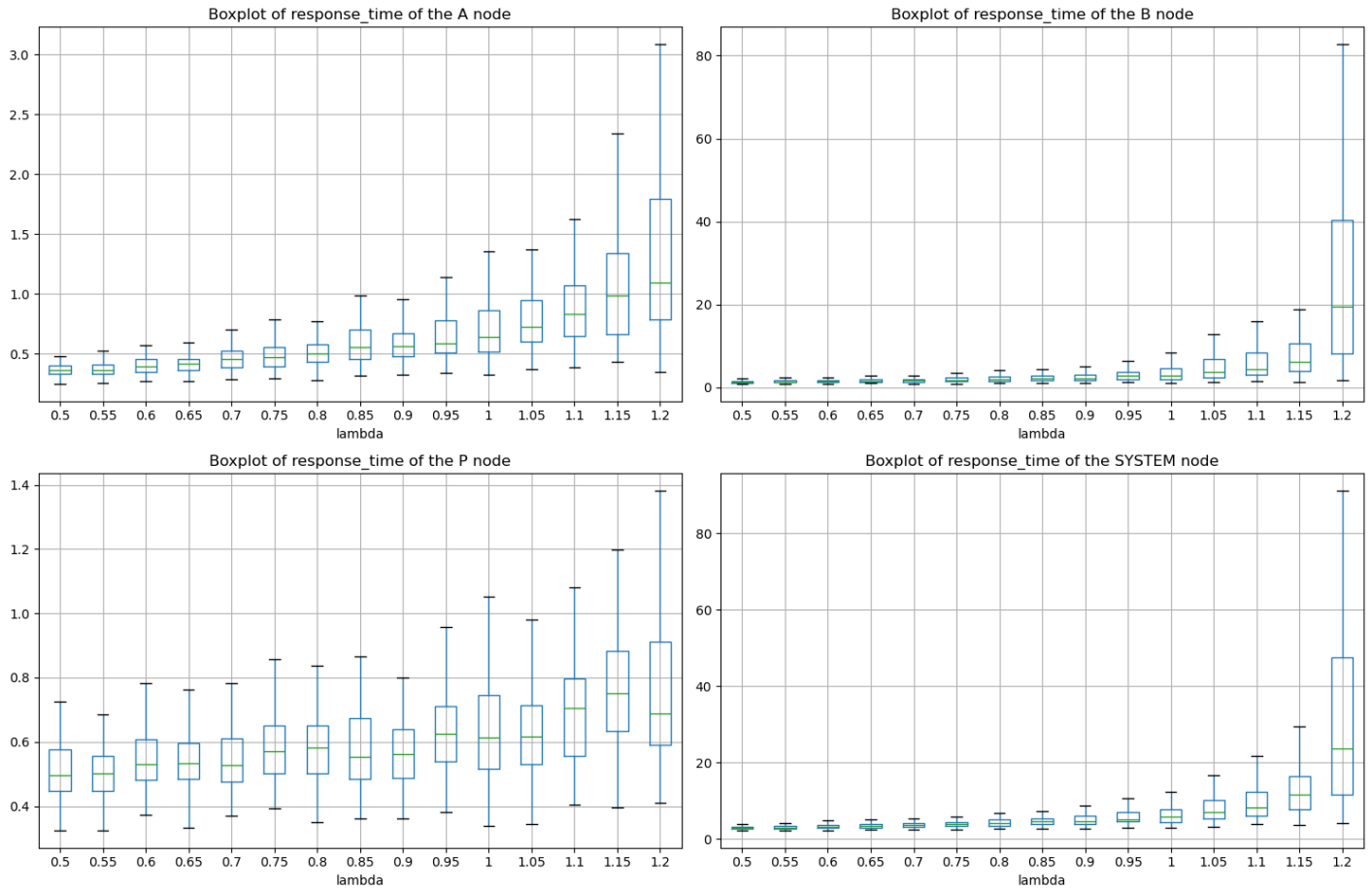


Figure 6: Distribuzione del Tempo di Risposta medio dei risultati sperimentali dell'obbiettivo 1

Objective 1: Finite Horizon

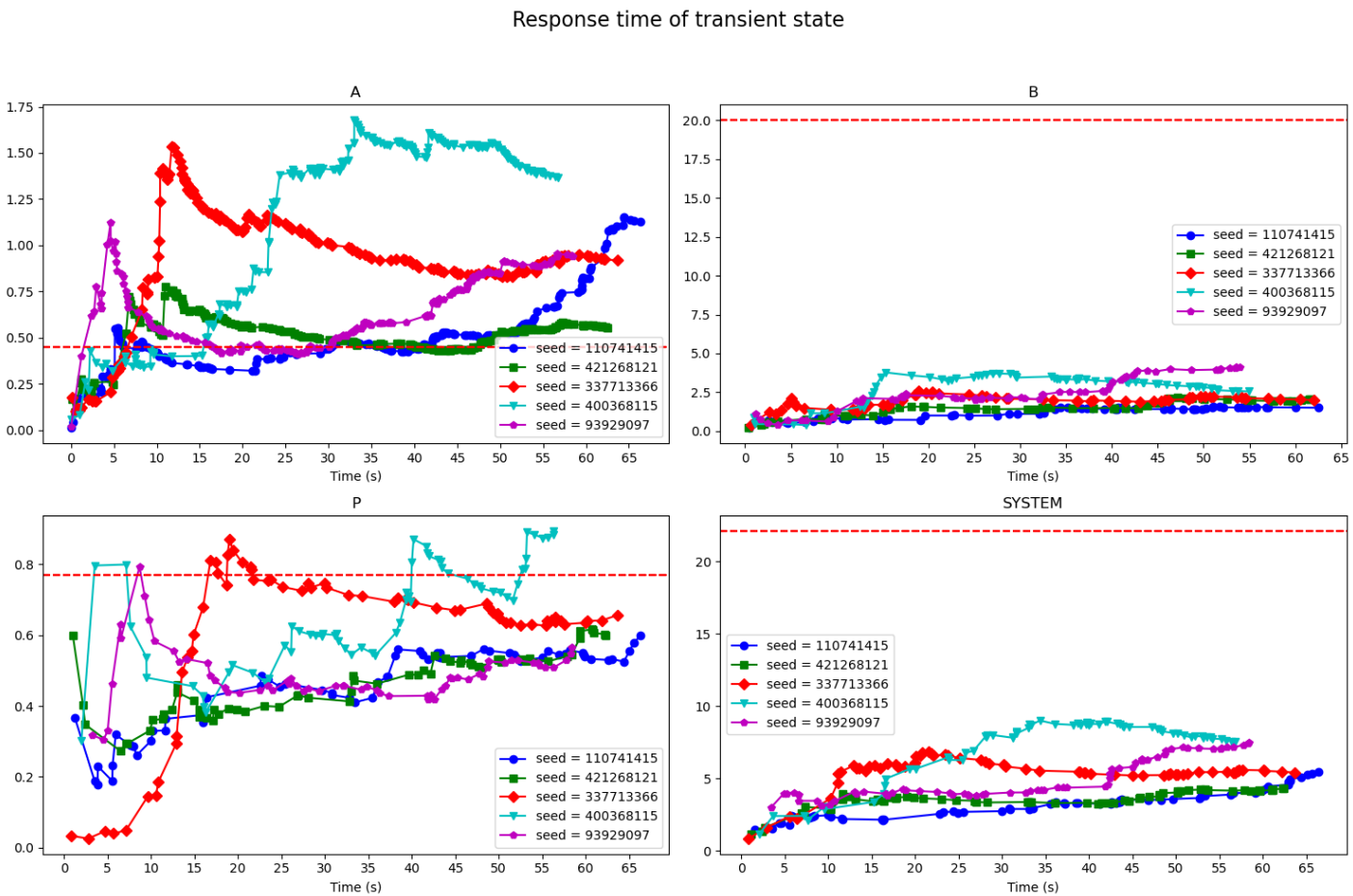


Figure 7: Tempo di risposta per l'obiettivo 1 in funzione del tempo di simulazione nello stato transiente del sistema con un rate di arrivi 1.2 job/s al variare del seed

Objective 1: Verification

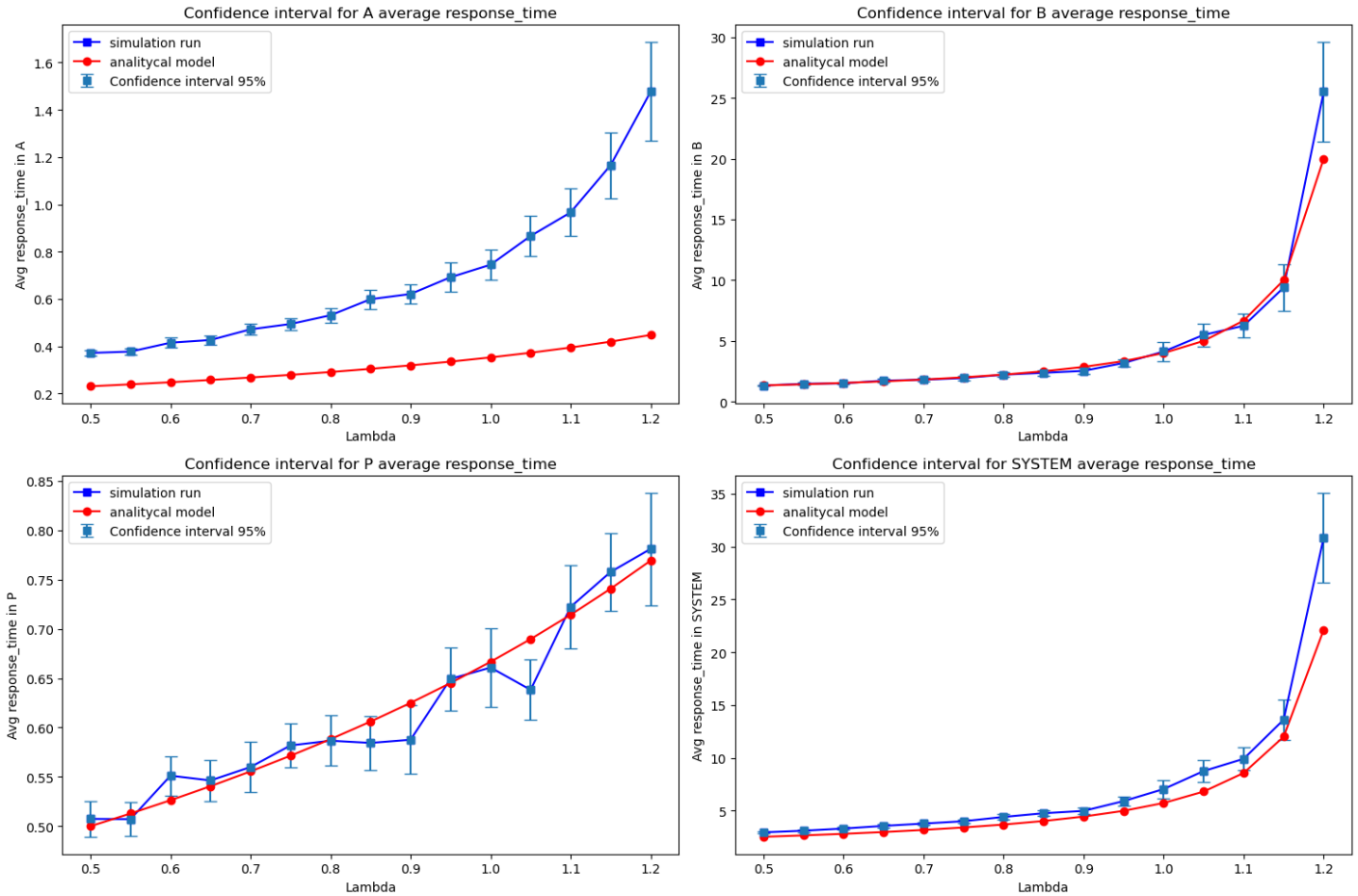


Figure 8: Confronto tra valori medi della simulazione e del modello analitico del Tempo di Risposta per l'Obiettivo 1.

Objective 2: Infinite Horizon

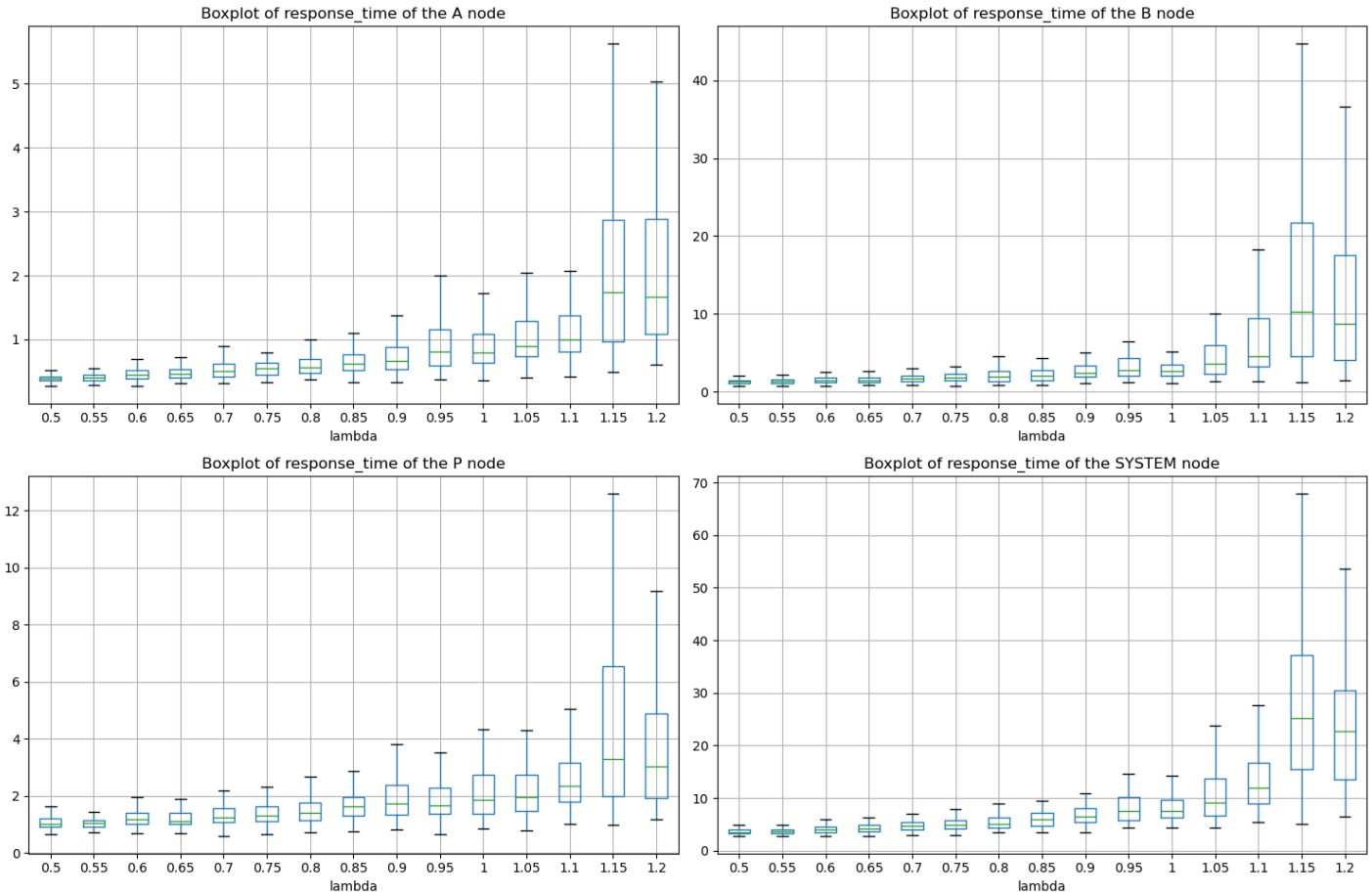


Figure 9: Distribuzione del Tempo di Risposta medio dei risultati sperimentali dell'obbiettivo 2

Objective 2: Finite Horizon

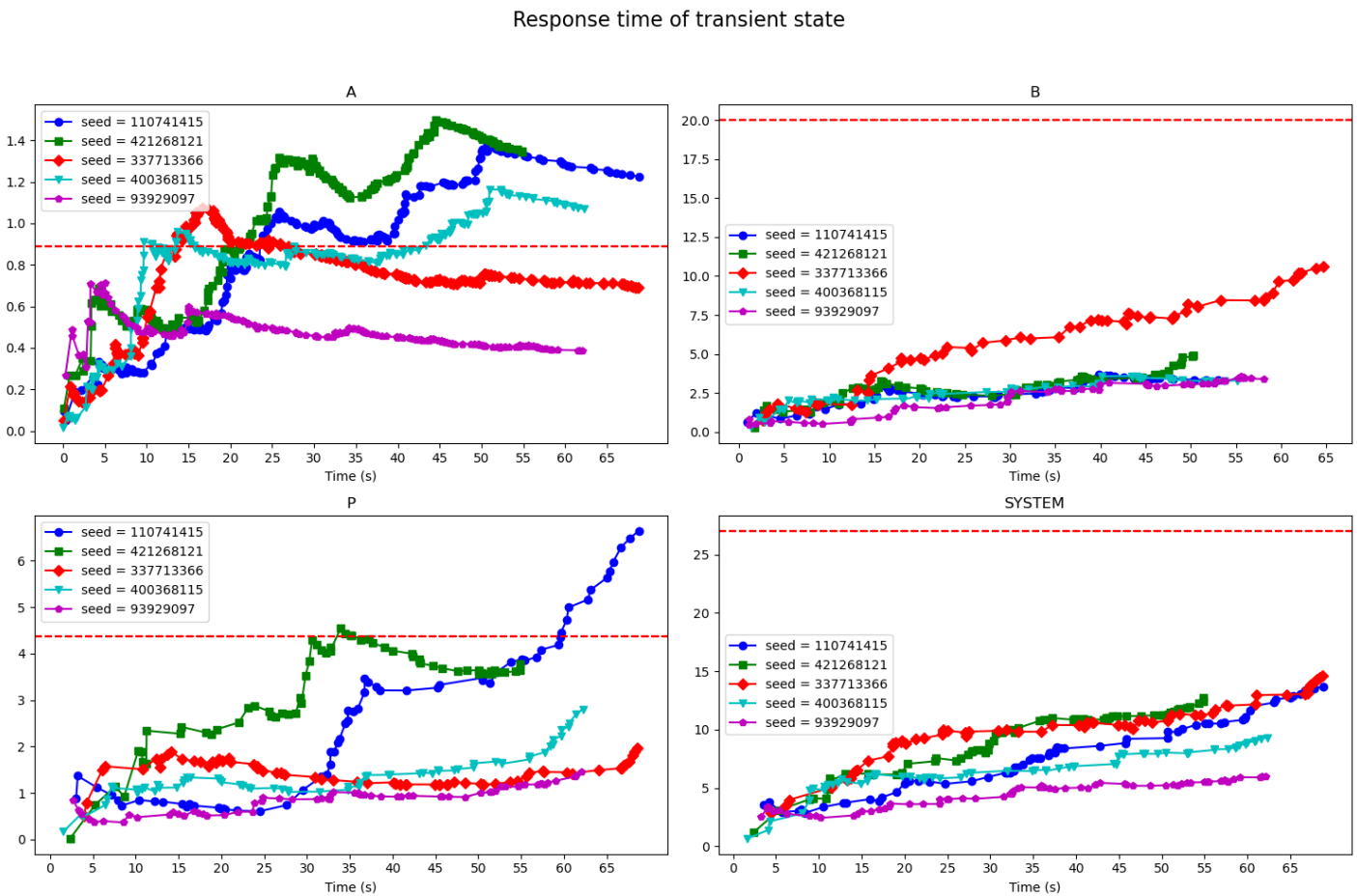


Figure 10: Tempo di risposta per l'obiettivo 2 in funzione del tempo di simulazione nello stato transiente del sistema con un rate di arrivi 1.2 job/s al variare del seed

Objective 2: Verification

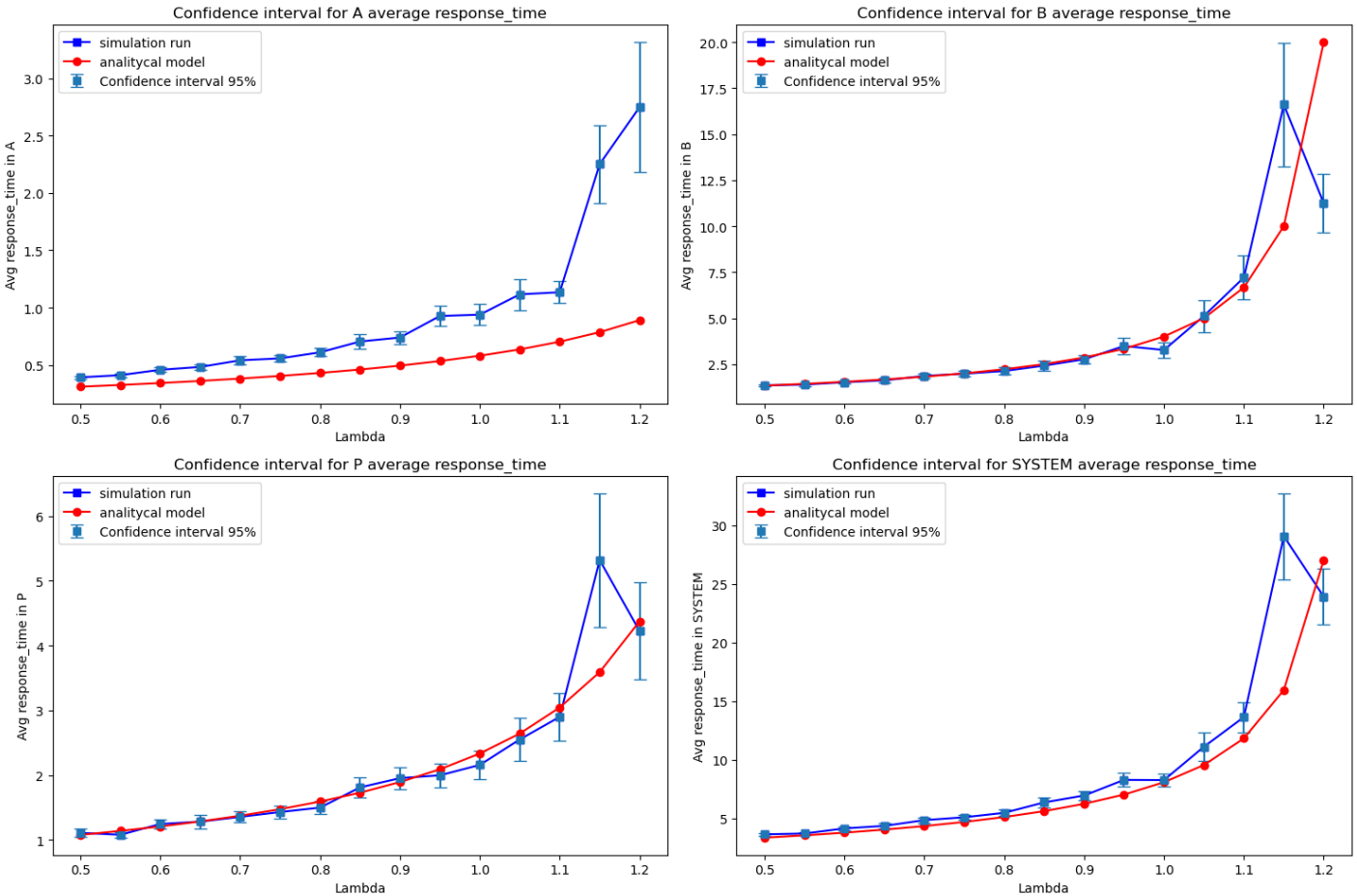
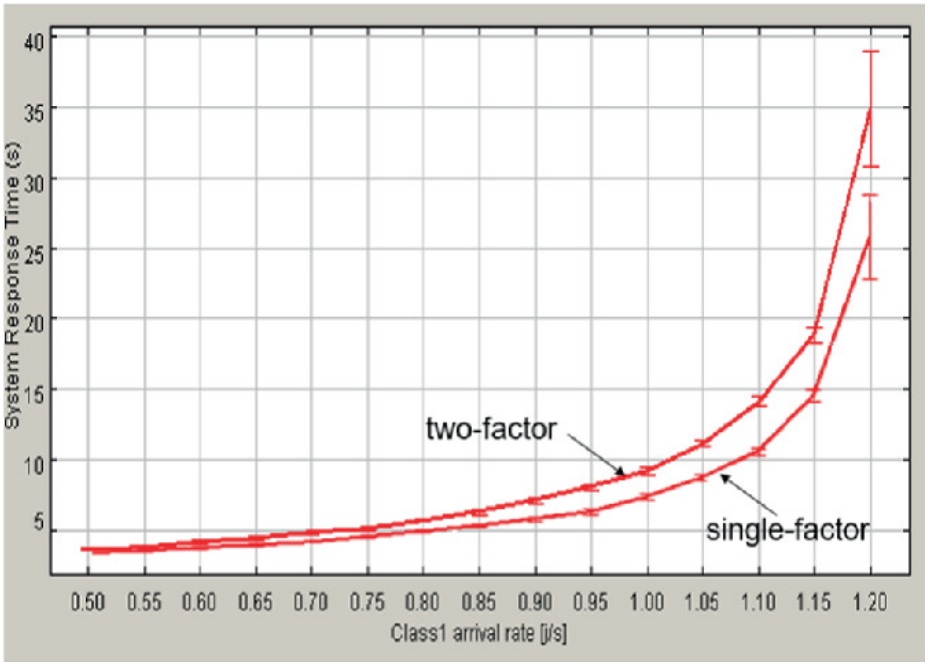
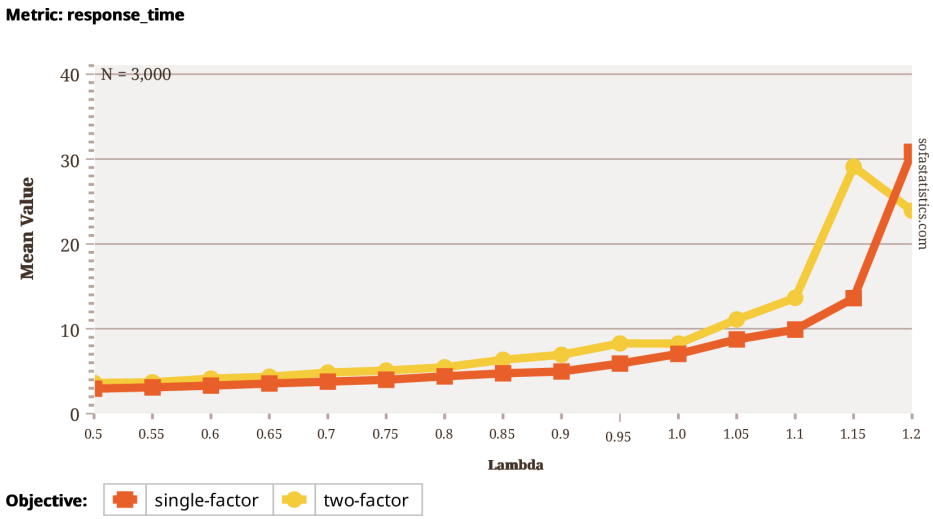


Figure 11: Confronto tra valori medi della simulazione e del modello analitico del Tempo di Risposta per l'Obiettivo 2.

Objective 2: Validation



(a) Case Study (Serazzi, 2024)



(b) Simulazione

Figure 12: Tempo di Risposta medi del sistema in funzione del rate di arrivi esterni nel caso con e senza 2FA.

Objective 3: Infinite Horizon

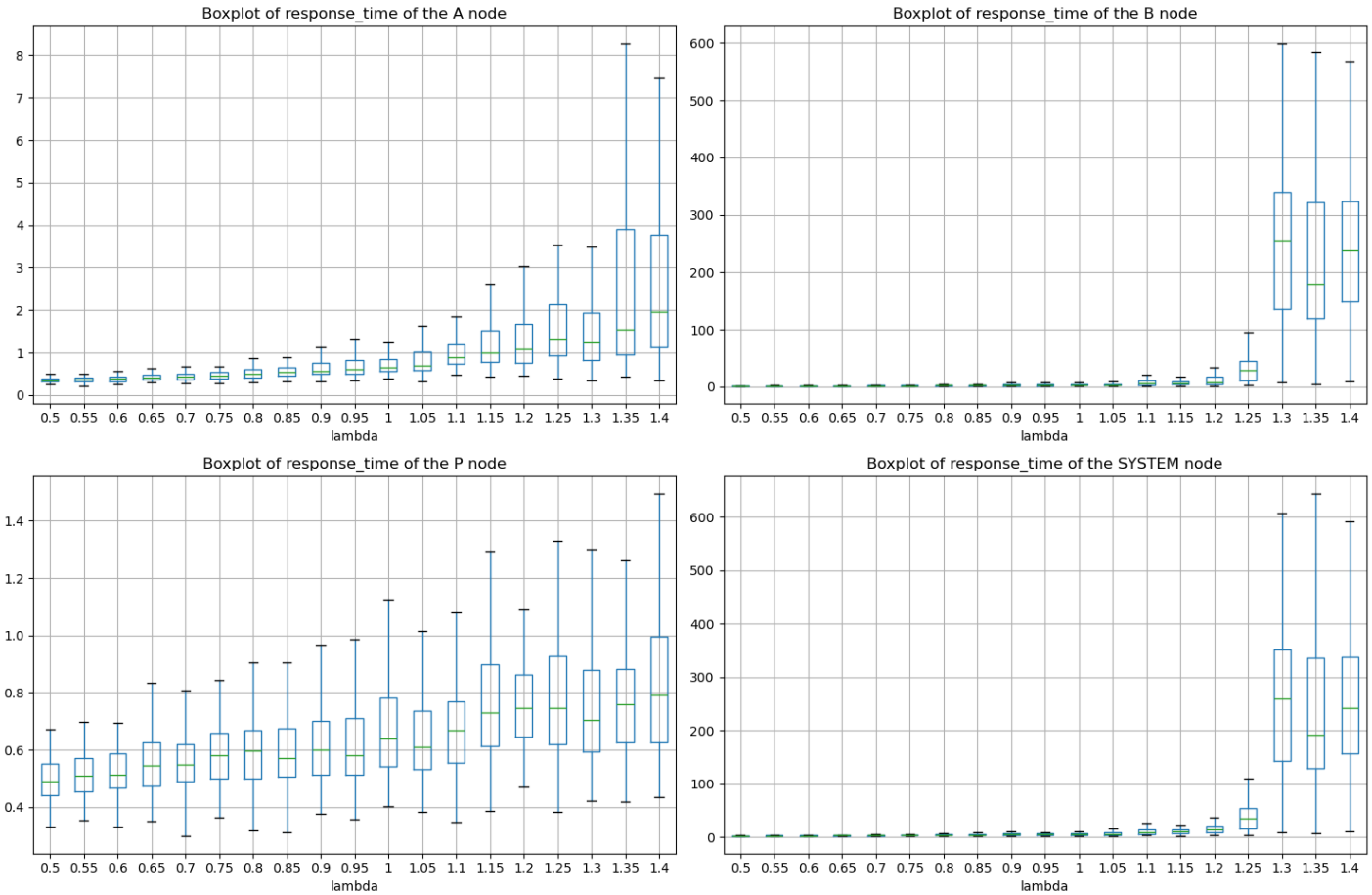


Figure 13: Distribuzione del Tempo di Risposta medio dei risultati sperimentali dell'obbiettivo 3

Objective 3: Finite Horizon

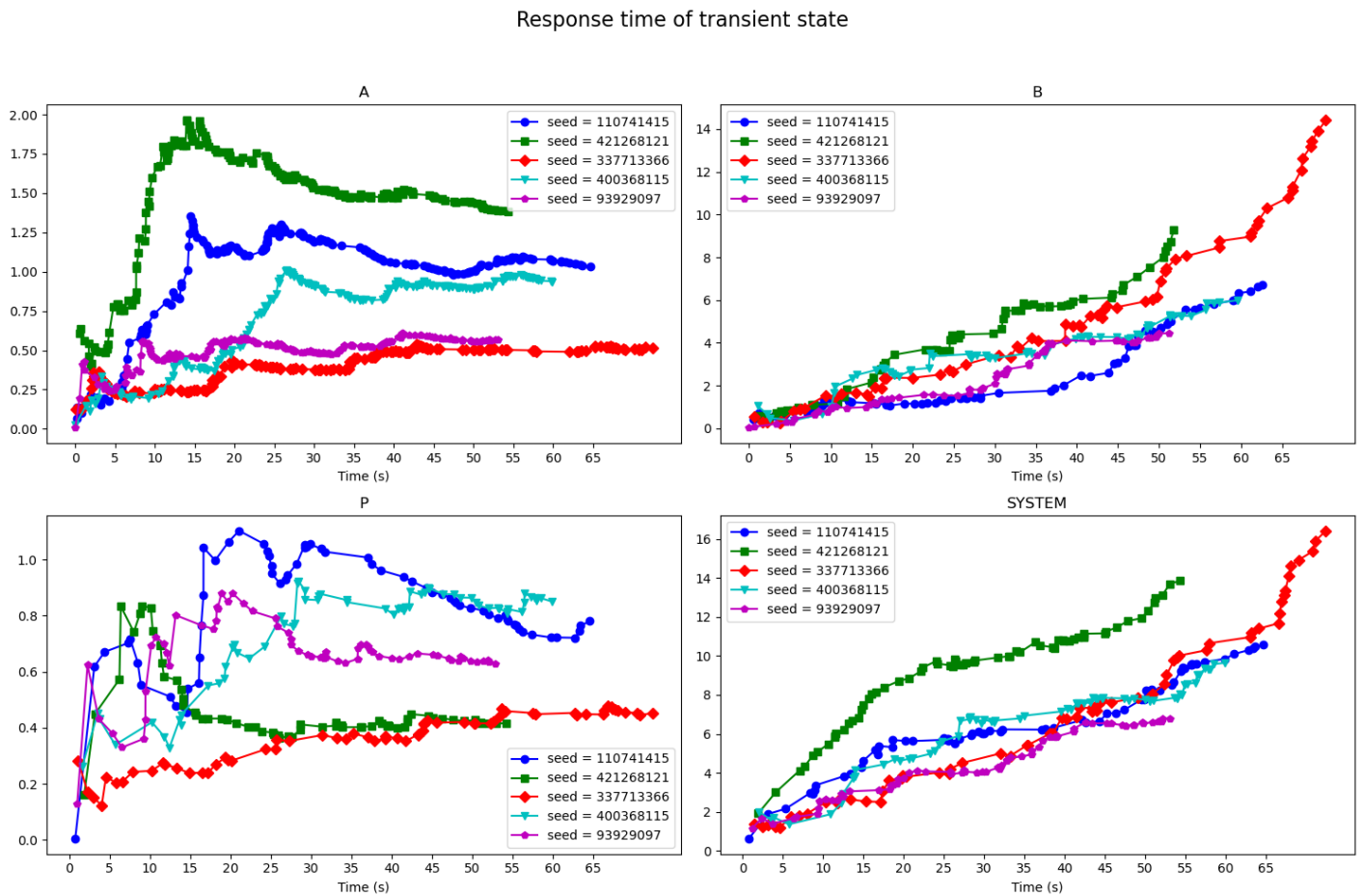


Figure 14: Tempo di risposta per l'obiettivo 3 in funzione del tempo di simulazione nello stato transiente del sistema con un rate di arrivi 1.4 job/s al variare del seed

Objective 3: Verification

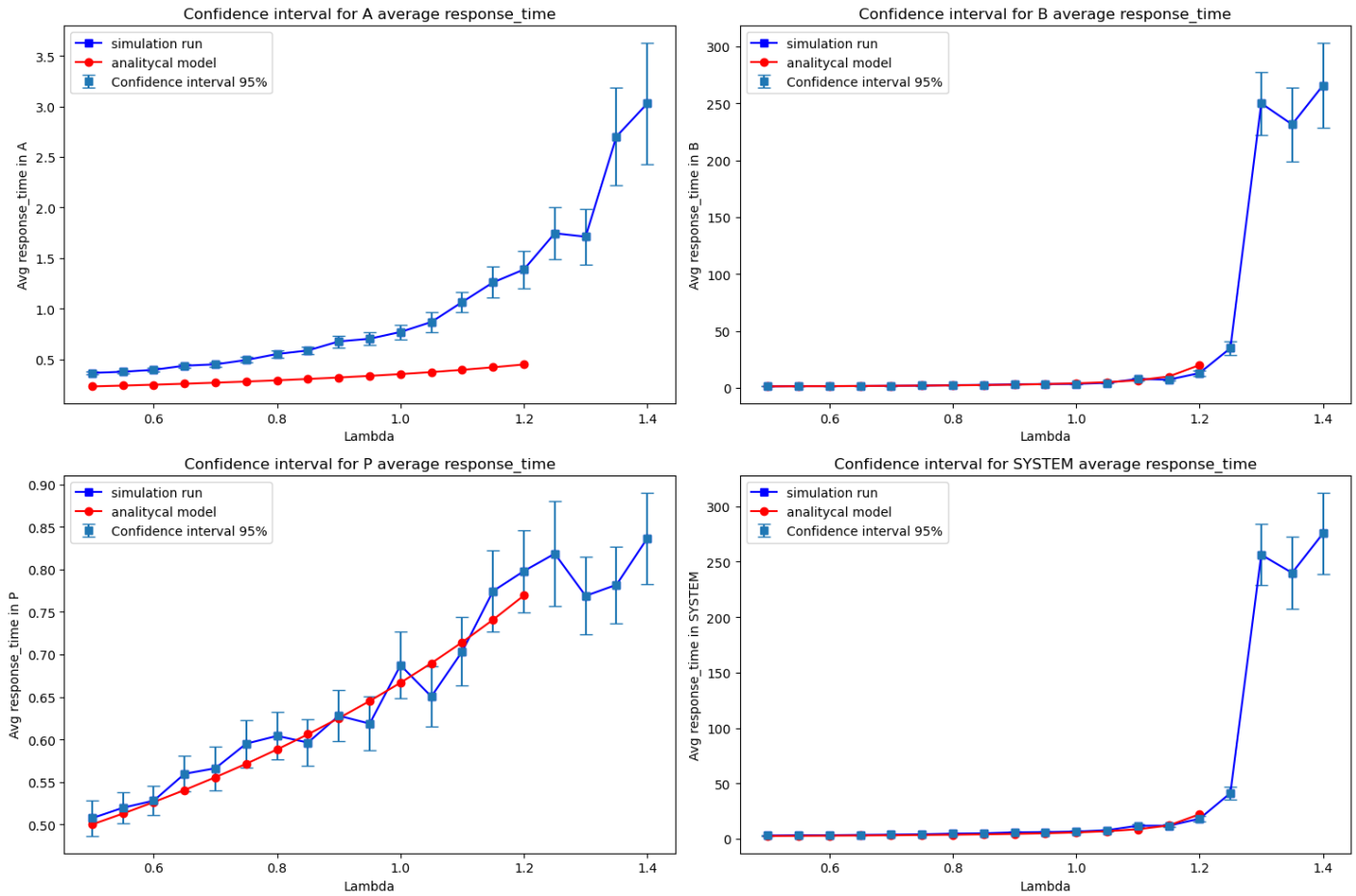


Figure 15: Confronto tra valori medi della simulazione e del modello analitico dell Tempo di Risposta per l'Obiettivo 3.

Objective 4: Throughput comparison

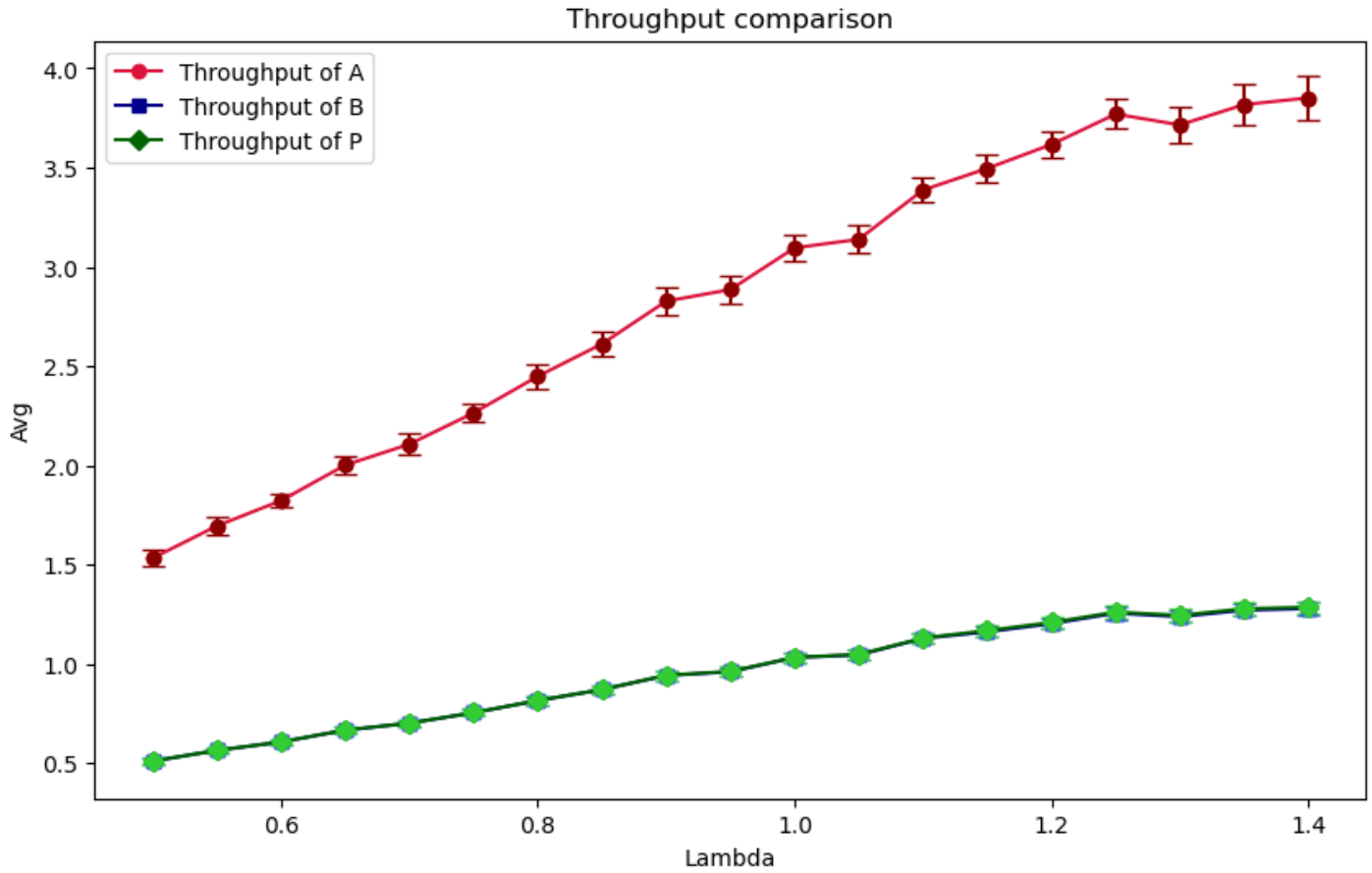
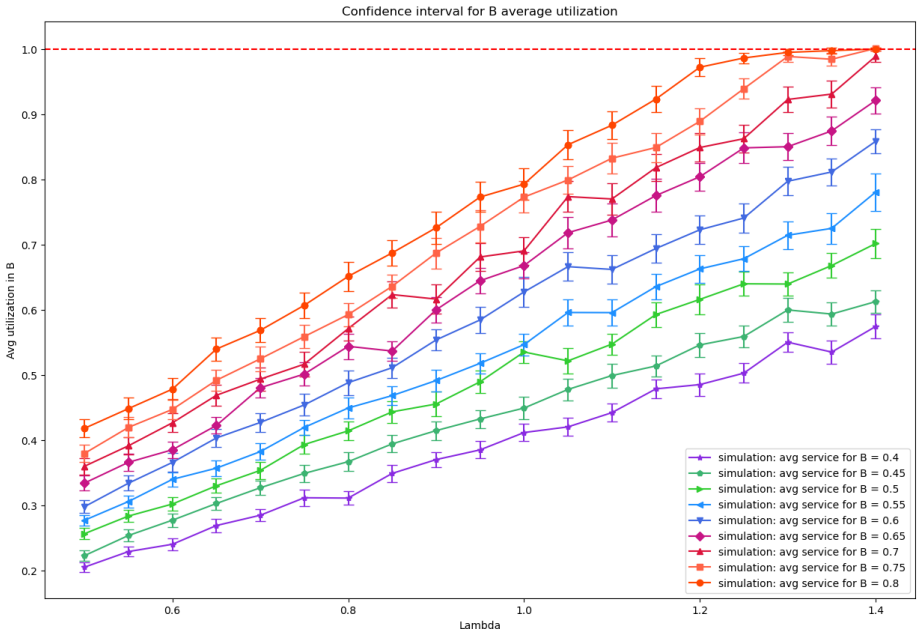
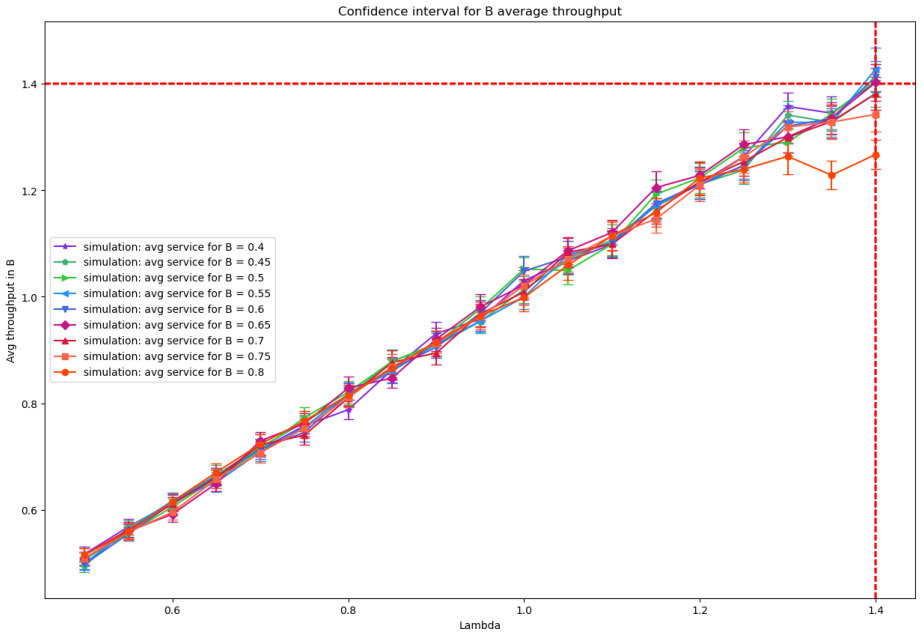


Figure 16: Confronto tra throughput dei tre server con tasso di arrivo $1.4 job/s$

Objective 4: Service time from 0.8 to 0.4



(a) Utilizzazione



(b) Throughput

Figure 17: Confronto Throughput e Utilizzazione medi al variare dei valori dei tempi di servizio

Objective 4: Service time 0.4 vs 0.65

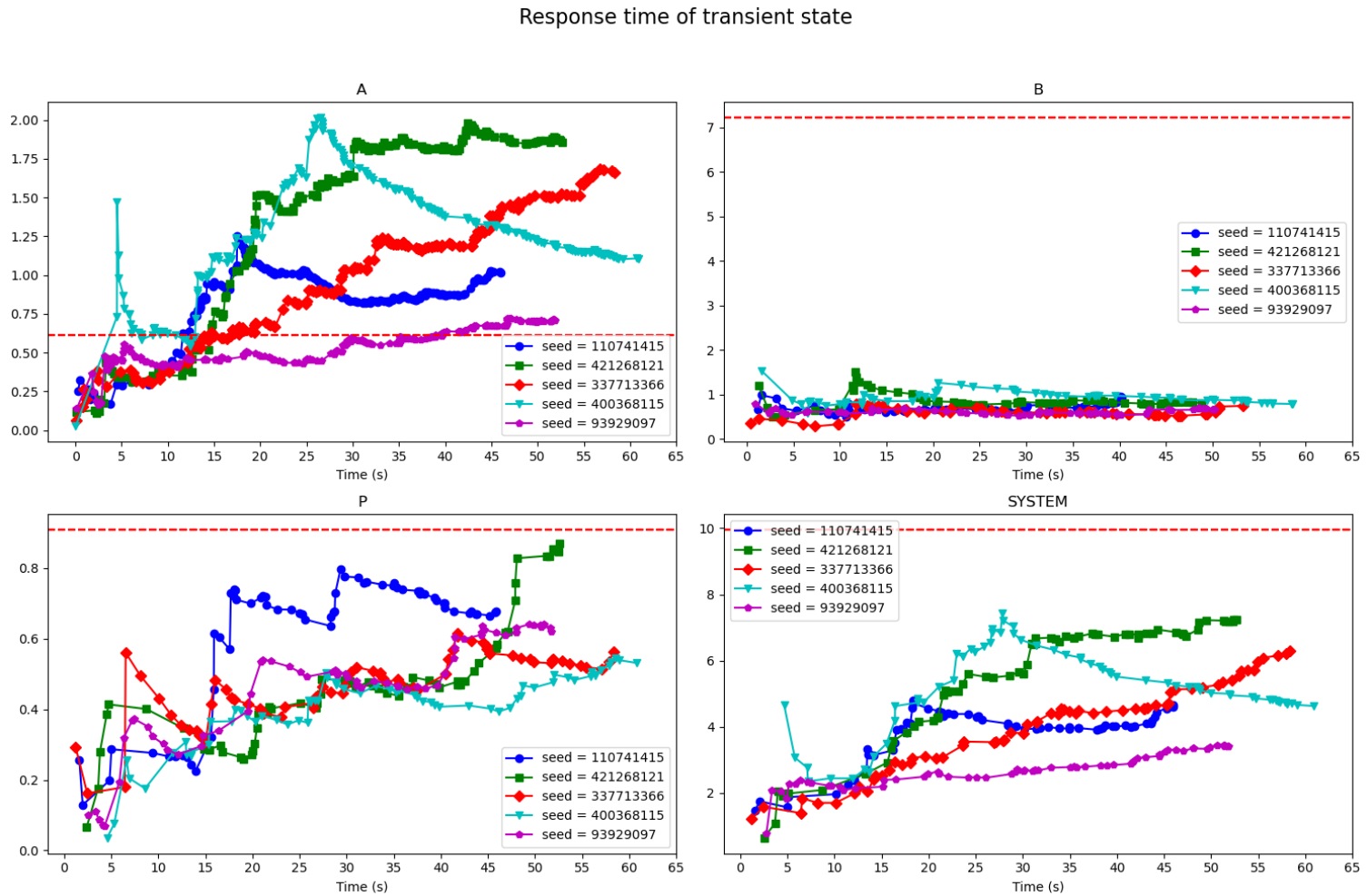
Table 1: Minimi e massimi di utilizzazione e throughput per tempi di servizio 0.65 del server B

λ	metrica	valore
0.5	utilizzazione	0.333898
1.4	utilizzazione	0.921304
0.5	throughput	0.507752
1.4	throughput	1.40273

Table 2: Minimi e massimi di utilizzazione e throughput per tempi di servizio 0.4 del server B

λ	metrica	valore
0.5	utilizzazione	0.20484
1.4	utilizzazione	0.574275
0.5	throughput	0.516347
1.4	throughput	1.40302

Objective 4: Finite Horizion (service time 0.4)



(a) Senza valore analitico

Response time of transient state

Objective 4: Verification (service time 0.4)

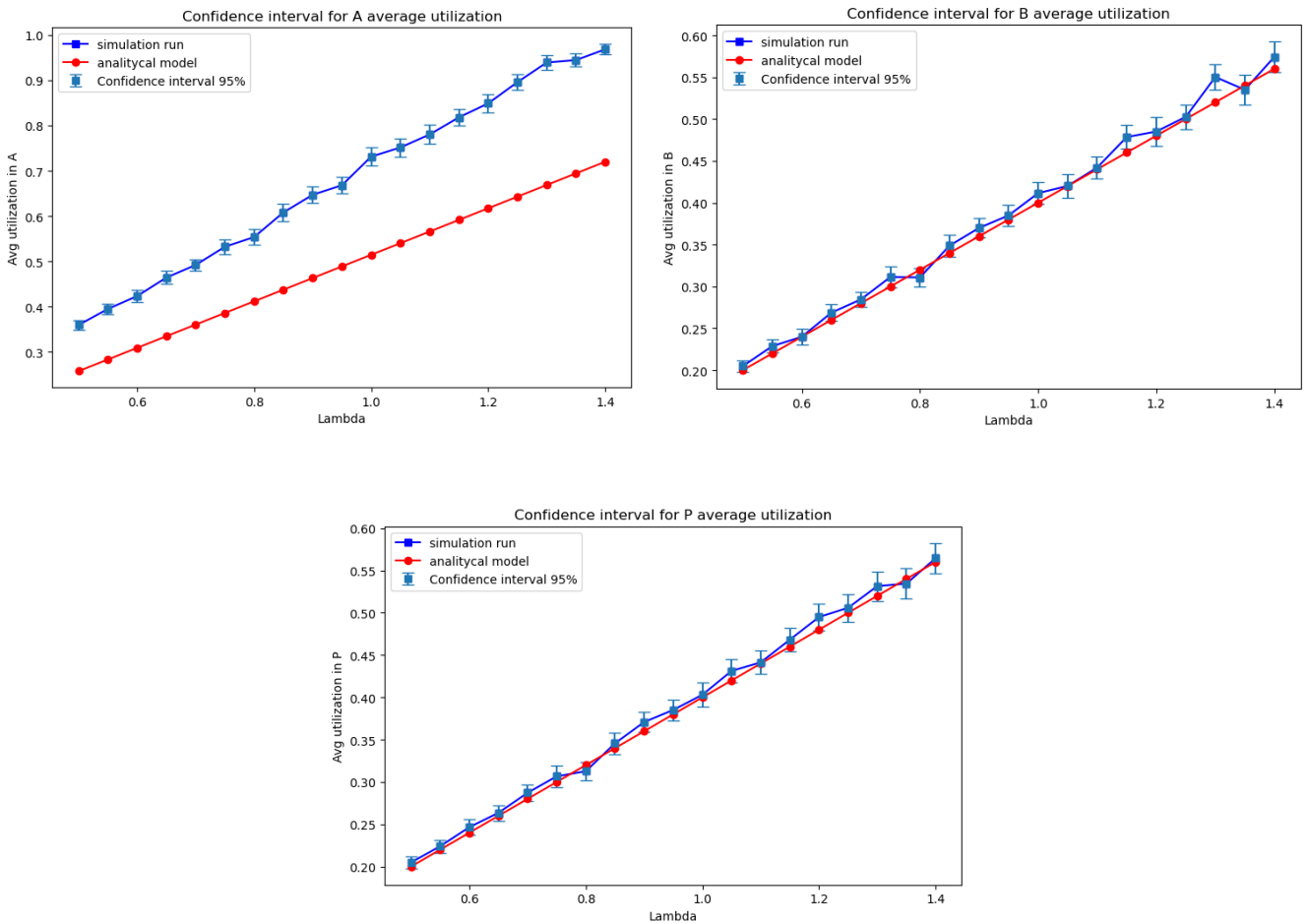
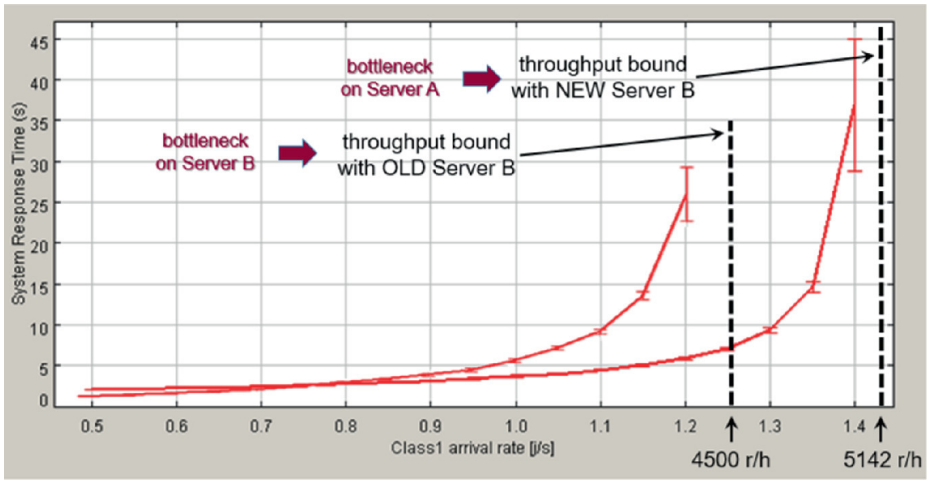
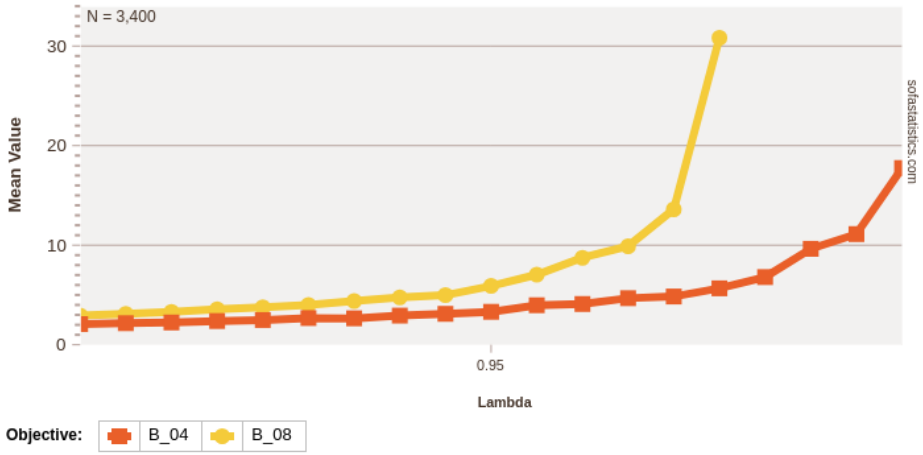


Figure 19: Intervallo di confidenza dell'utilizzazione e confronto con modello analitico per l'obiettivo 4.

Objective 4: Validation



(a) Caso di studio (Serazzi, 2024)



(b) Risultati sperimentali

Figure 20: Tempo di risposta medio prima e dopo la miglioria del server B in funzione del rate medio degli arrivi esterni

References

Stephen K. Park Lawrence M. Leemis. 2004. *Discrete-Event Simulation: A First Course*. Prentice Hall. https://www.math.wm.edu/~leemis/

Giuseppe Serazzi. 2024. *Performance Engineering - Learning Through Applications Using JMT*. Springer. https://doi.org/10.1007/978-3-031-36763-2