

Gestione settore stadio

CRECCO VALERIO 0320452

SPAZIANI PIERPAOLO 0316331

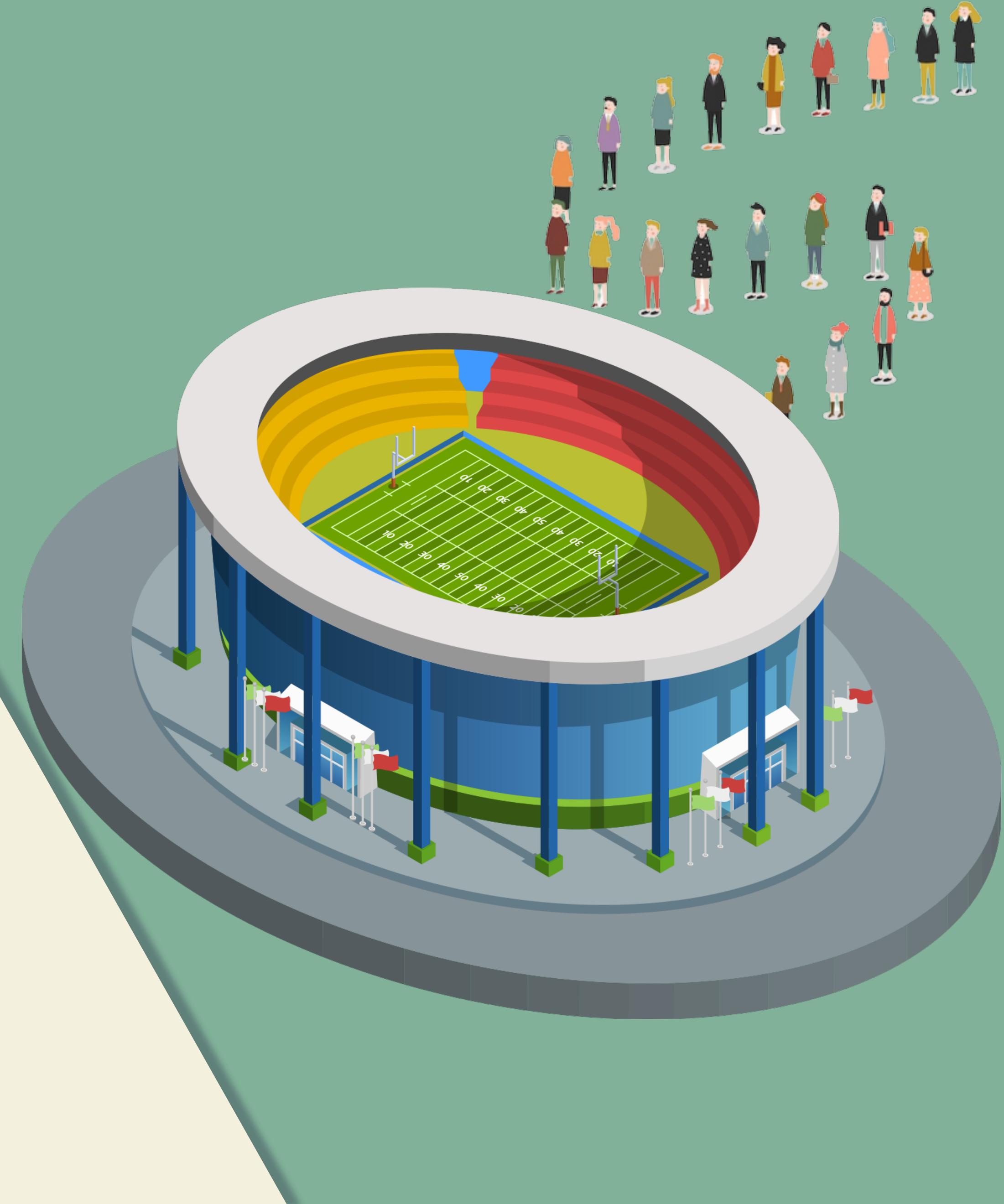
DE SANTIS LUDOVICO 0320460

INTRODUZIONE

Prendendo spunto dalla nostra esperienza personale in ambito lavorativo, è stato deciso di modellare la gestione del flusso di arrivo in un settore dello stadio.

I problemi solitamente riscontrati sono:

- ▶ lunghe attese in coda
- ▶ scarsi controlli
- ▶ costi elevati per il personale



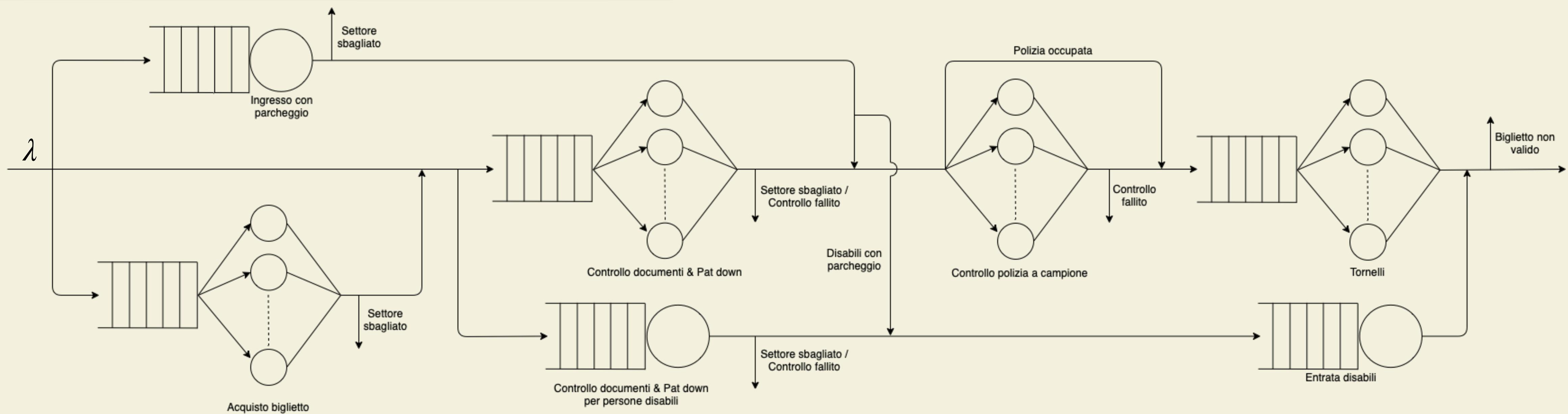
OBIETTIVI

Gli obiettivi preposti sono:

- ▶ minimizzare i costi
- ▶ rispettare i QoS:
 - avere tempi massimi di risposta del sistema di **6 minuti e mezzo**
 - avere una copertura di controlli della polizia almeno del **25%**



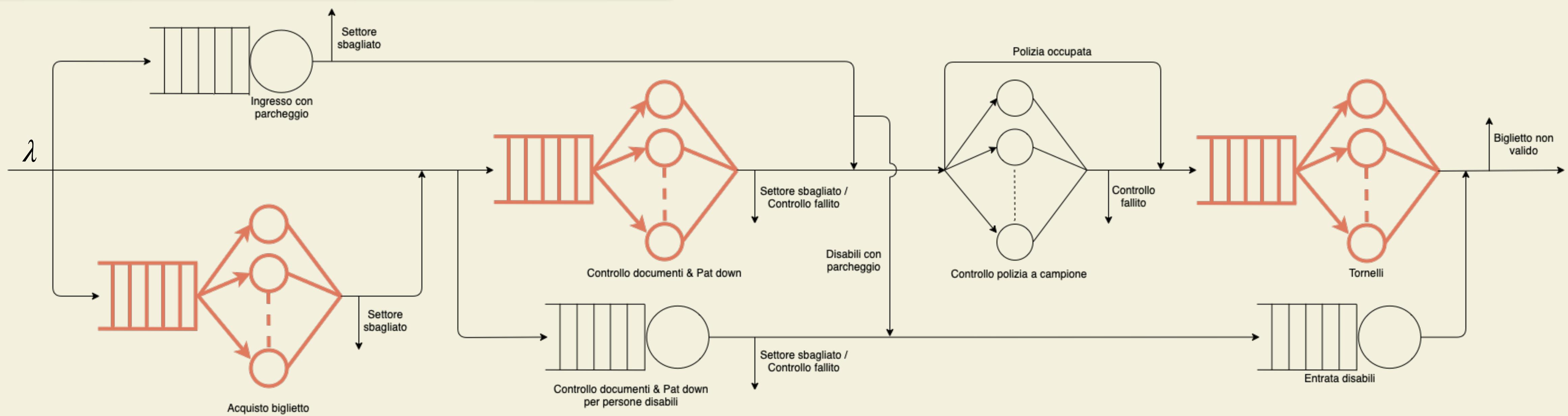
MODELLO CONCETTUALE



Centri modellati

- ▶ Park Entry
- ▶ Ticket Buy
- ▶ Police Control
- ▶ Doc & Pat
- ▶ Doc & Pat D.
- ▶ Turnstiles
- ▶ Turnstiles D.

MODELLO CONCETTUALE

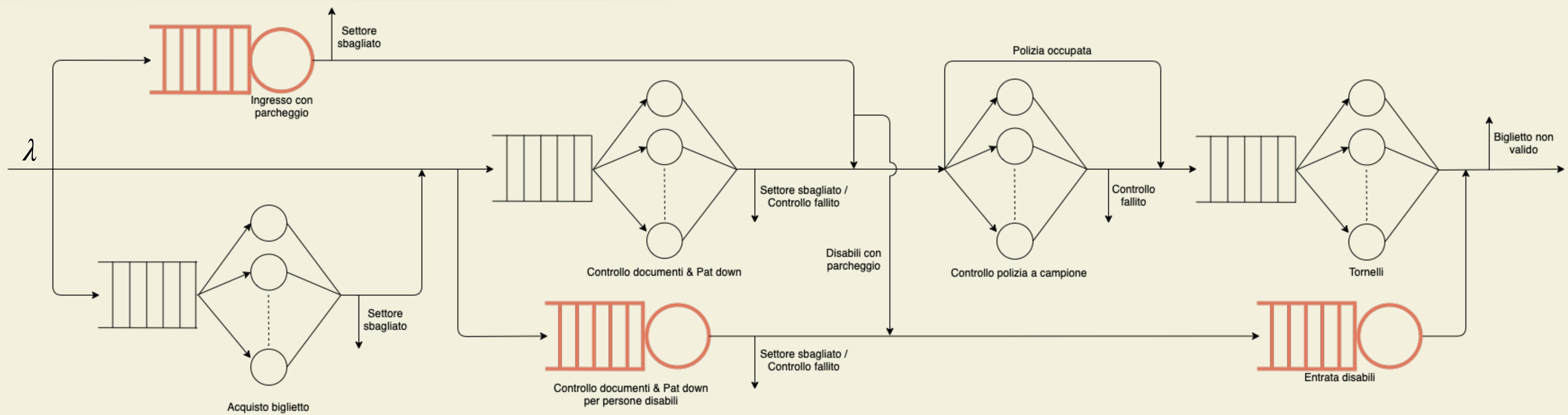


Ticket Buy – Doc & Pat – Turnstiles

Modellati con una **M/M/m**:

- ▶ Code infinite
- ▶ Possibilità di gestire più utenti in parallelo

MODELLO CONCETTUALE

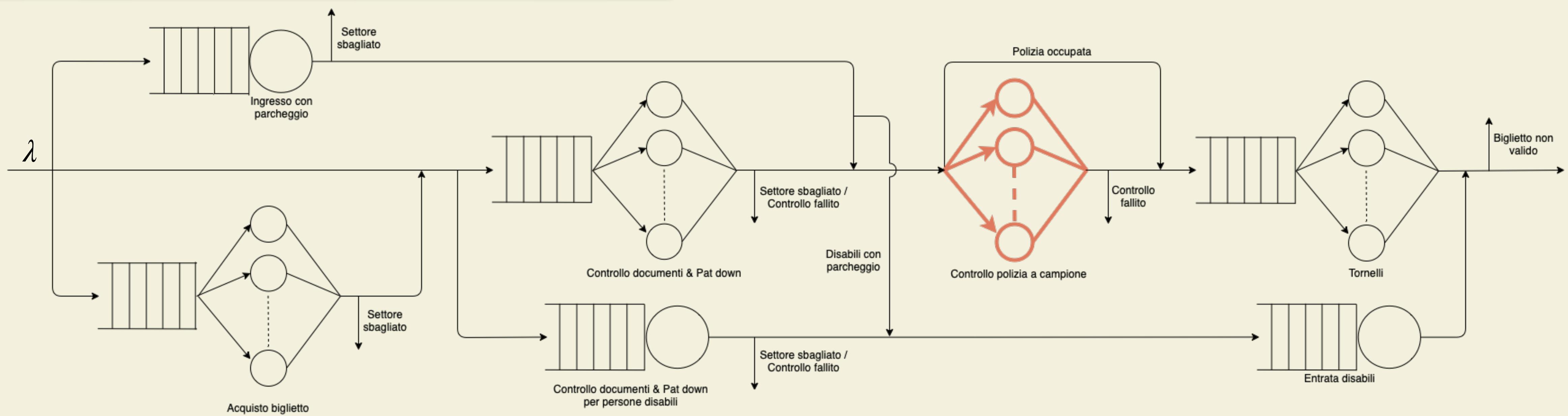


Park Entry – Doc & Pat D. – Turnstiles D.

Modellati con una **M/M/1**:

- ▶ Code infinite
- ▶ Singolo servente, si prevede un affluenza minore

MODELLO CONCETTUALE



Police Control

Modellato con una **M/M/m/m**:

- ▶ Nessuna coda (servizio a capacità limitata)
- ▶ Possibilità di gestire più utenti in parallelo

MODELLO CONCETTUALE

Gli **utenti** presenti sono:

- ▶ utenti senza disabilità
- ▶ utenti con disabilità

Gli **eventi** che possono verificarsi sono:

- ▶ arrivo di un utente
- ▶ completamento di una richiesta



VARIABILI DI STATO

Le **variabili di stato** sono:

- ▶ numero di serventi disponibili in ogni fascia oraria
- ▶ numero di utenti presenti
- ▶ numero di utenti in servizio
- ▶ rapporto controlli su arrivi nel centro *Police Control*



MODELLO DELLE SPECIFICHE

Il periodo di osservazione considerato è di **3 ore**.

Assumendo che una partita abbia inizio alle ore 15:00 e che i cancelli vengono aperti due ore prima, sono state individuate **tre fasce orarie** con differenti flussi di arrivo:

- ▶ **13:00-14:00:** flusso medio (30%) → $\lambda_1 = 0.5875$ persone/sec
- ▶ **14:00-15:00:** flusso alto (65%) → $\lambda_2 = 1.273056$ persone/sec
- ▶ **15:00-16:00:** flusso basso (5%) → $\lambda_3 = 0.09778$ persone/sec

I tempi di **interarrivo** e **servizio** sono stati modellati tramite **distribuzioni esponenziali**.

* è stato considerato il 94% della capienza della curva dello Stadio Olimpico di Roma

MODELLO DELLE SPECIFICHE

| GRANDEZZE | VALORI |
|--|----------------|
| Numero medio di tifosi per un settore specifico | 7050* |
| Tempo medio servizio ingresso con parcheggio | 30 sec/persona |
| Tempo medio servizio acquisto biglietto | 90 sec/persona |
| Tempo medio servizio controllo documenti e pat down | 20 sec/persona |
| Tempo medio servizio controllo documenti e pat down per disabili | 20 sec/persona |

MODELLO DELLE SPECIFICHE

| GRANDEZZE | VALORI |
|---|----------------|
| Tempo medio servizio controllo polizia a campione | 40 sec/persona |
| Tempo medio servizio tornelli | 15 sec/persona |
| Tempo medio servizio entrata disabili | 15 sec/persona |
| Costo medio staff | 12.5 euro/ora |
| Costo medio poliziotti | 23.5 euro/ora |

MODELLO DELLE SPECIFICHE

| GRANDEZZE | VALORI |
|--|------------------|
| Percentuale arrivi ingresso con parcheggio | 2% ~ 141 persone |
| Percentuale acquisti biglietti in loco | 8% ~ 564 persone |
| Percentuale disabili senza parcheggio | 1% ~ 67 persone |
| Percentuale disabili con parcheggio | 65% ~ 92 persone |
| Percentuale settore sbagliato parcheggio | 1% ~ 2 persone |

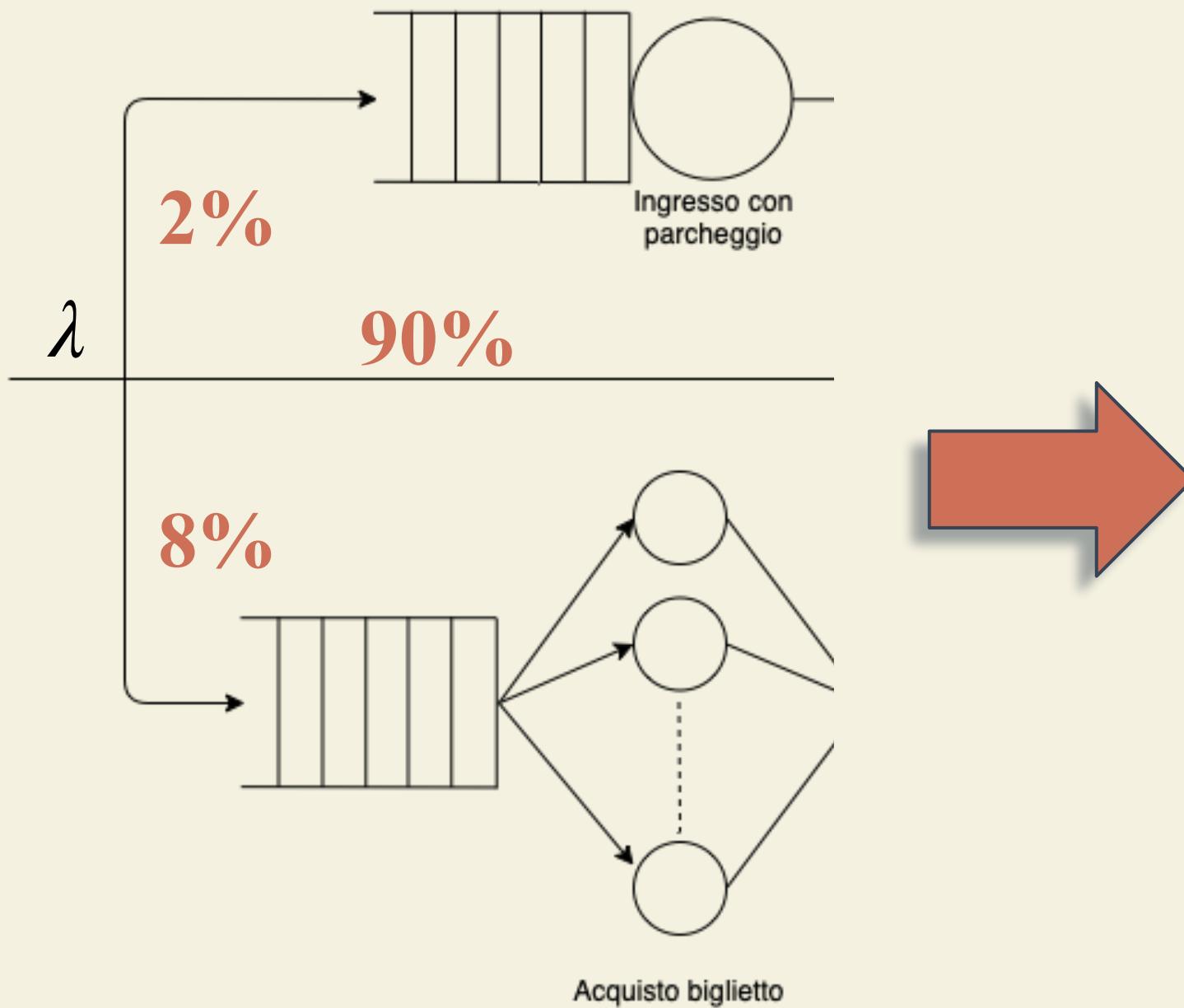
MODELLO DELLE SPECIFICHE

| GRANDEZZE | VALORI |
|---|--------------------|
| Percentuale settore sbagliato acquisto biglietti | 15% ~ 85 persone |
| Percentuale settore sbagliato controllo documenti e pat down | 5% ~ 338 persone |
| Percentuale settore sbagliato controllo documenti e pat down per persone disabili | 1% ~ 1 persone |
| Percentuale persone controllo polizia fallito | 5% ~ 81 persone |
| Percentuale biglietto sbagliato | 0.25% ~ 17 persone |

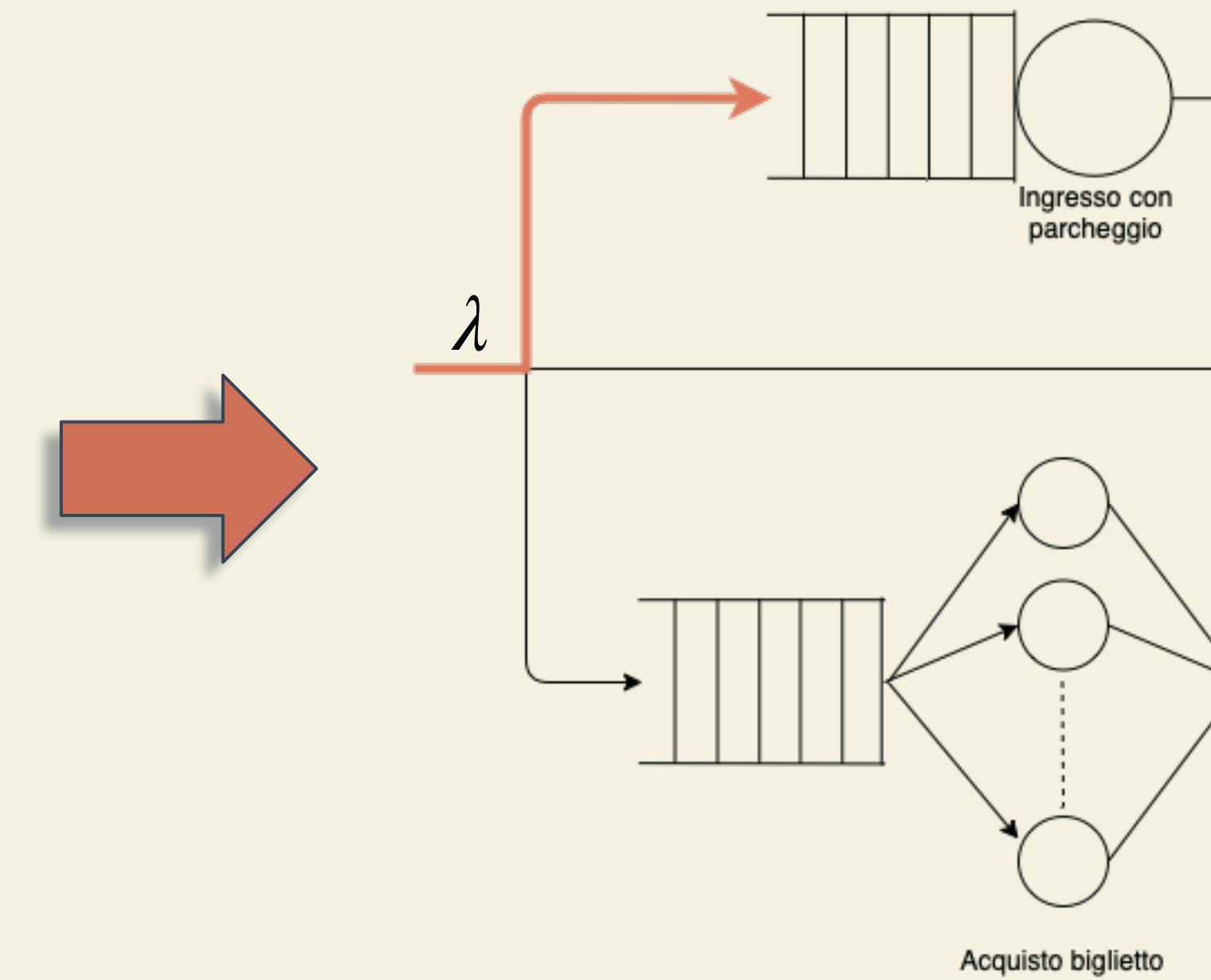
MODELLO COMPUTAZIONALE

Nei centri **multi-server** è stato utilizzato un approccio *Equity*: ogni utente prende servizio sul servente che risulta libero da più tempo.

Al fine di realizzare il corretto instradamento degli utenti, sono state rispettate le **probabilità di routing** indicate nel modello delle specifiche, utilizzando la funzione *Uniform(0, 1)* presente nel file rvgs.c.



```
x = Uniform(0,1);
if (x <= 0.02) {
    routing PARK_ENTRY;
} else if (x <= 0.10) {
    routing TICKET_BUY;
} else {
    routing ... ;
}
```



VERIFICA

Per assicurare che i risultati siano conformi alle specifiche, sono stati osservati i seguenti criteri:

- ▶ Il numero di utenti in un centro deve essere uguale al numero di utenti in coda più il numero di utenti in servizio
- ▶ Il tempo di risposta deve essere uguale alla somma del tempo trascorso in coda più il tempo di servizio
- ▶ Il numero di arrivi deve essere uguale al numero di completamenti più il numero di *dropped*
- ▶ Il numero di utenti in ingresso in un centro è determinato dalla probabilità di routing indicata nel modello delle specifiche

* la simulazione riportata fa riferimento alla fascia oraria 14:00 - 15:00 ($\lambda_2 = 1.273056$)

VERIFICA

Park Entry (SSQ - M/M/1)

- ▶ $E(N_S) = E(N_Q) + \rho =$
 $= 3.24910 + 0.82008 = 4.06918$
- ▶ $E(T_S) = E(T_Q) + E(S) =$
 $= 107.02838 + 27.01403 = 134.04241$
- ▶ Arrivi = Completamenti + Dropped =
 $= 114 + 1 = 115$
- ▶ #expected_arrivals = $\lambda_2 \cdot 3600 \cdot p_{PE} =$
 $= 1.273056 \cdot 3600 \cdot 0.02 = 91.66003$

```
for 114 jobs the service node statistics:  
# arrivals ..... = 115  
# dropped ..... = 1  
# completions ..... = 114  
avg interarrivals .. = 31.29885  
avg # in node ..... = 4.06918  
avg # in queue ..... = 3.24910  
avg utilization .... = 0.82008  
avg wait ..... = 134.04241  
avg delay ..... = 107.02838  
avg service ..... = 27.01403
```

* la simulazione riportata fa riferimento alla fascia oraria 14:00 - 15:00 ($\lambda_2 = 1.273056$)

VERIFICA

Park Entry (SSQ - M/M/1)

- ▶ Gli arrivi effettivi (115) risultano *outlier* rispetto a quelli teorici (91.66003).
- ▶ Effettuando la simulazione per 256 ripetizioni, i valori medi risultano:

| Arrivi | Dropped | Completamenti |
|----------|---------|---------------|
| 92.25781 | 0.94141 | 91.31641 |

```
for 114 jobs the service node statistics:  
# arrivals ..... = 115  
# dropped ..... = 1  
# completions ..... = 114  
avg interarrivals .. = 31.29885  
avg # in node ..... = 4.06918  
avg # in queue ..... = 3.24910  
avg utilization .... = 0.82008  
avg wait ..... = 134.04241  
avg delay ..... = 107.02838  
avg service ..... = 27.01403
```

VERIFICA

Ticket Buy (MSQ - M/M/11)

- ▶ $E(N_S) = E(N_Q) + m\rho =$
 $= 1.97292 + 11 \cdot 0.8003 = 10.77664$
- ▶ $E(T_S) = E(T_Q) + E(S) =$
 $= 20.85747 + 93.07180 = 113.92926$
- ▶ Arrivi = Completamenti + Dropped =
 $= 299 + 68 = 367$
- ▶ #expected_arrivals = $\lambda_2 \cdot 3600 \cdot p_{TB} =$
 $= 1.273056 \cdot 3600 \cdot 0.08 = 366.64013$

```
for 299 jobs the service node statistics:  
# arrivals ..... = 367  
# dropped ..... = 68  
# completions ..... = 299  
avg interarrivals .. = 9.79972  
avg # in node ..... = 10.77664  
avg # in queue ..... = 1.97292  
avg utilization .... = 0.80034  
avg wait ..... = 113.92926  
avg delay ..... = 20.85747  
avg service ..... = 93.07180
```

* la simulazione riportata fa riferimento alla fascia oraria 14:00 - 15:00 ($\lambda_2 = 1.273056$)

VERIFICA

Ticket Buy (MSQ - M/M/11)

- ▶ Gli arrivi effettivi (367) risultano *outlier* rispetto a quelli teorici (366.64013).
- ▶ Effettuando la simulazione per 256 ripetizioni, i valori medi risultano:

| Arrivi | Dropped | Completamenti |
|-----------|----------|---------------|
| 366.89844 | 54.45703 | 312,44141 |

```
for 299 jobs the service node statistics:  
# arrivals ..... = 367  
# dropped ..... = 68  
# completions ..... = 299  
avg interarrivals .. = 9.79972  
avg # in node ..... = 10.77664  
avg # in queue ..... = 1.97292  
avg utilization .... = 0.80034  
avg wait ..... = 113.92926  
avg delay ..... = 20.85747  
avg service ..... = 93.07180
```

* la simulazione riportata fa riferimento alla fascia oraria 14:00 - 15:00 ($\lambda_2 = 1.273056$)

VERIFICA

Doc & Pat (MSQ - M/M/26)

- ▶ $E(N_S) = E(N_Q) + m\rho =$
 $= 3.69777 + 26 \cdot 0.87 = 26.31702$
- ▶ $E(T_S) = E(T_Q) + E(S) =$
 $= 3.26183 + 19.95260 = 23.21443$
- ▶ Arrivi = Completamenti + Dropped =
 $= 4094 + 213 = 4307$
- ▶ #expected_arrivals = 4391.98056 **
** il calcolo è stato effettuato con lo script 'population.py'

```
for 4094 jobs the service node statistics:  
# arrivals ..... = 4307  
# dropped ..... = 213  
# completions ..... = 4094  
avg interarrivals .. = 0.83595  
avg # in node ..... = 26.31702  
avg # in queue ..... = 3.69777  
avg utilization .... = 0.86997  
avg wait ..... = 23.21443  
avg delay ..... = 3.26183  
avg service ..... = 19.95260
```

* la simulazione riportata fa riferimento alla fascia oraria 14:00 - 15:00 ($\lambda_2 = 1.273056$)

VERIFICA

Doc & Pat (MSQ - M/M/26)

- ▶ Gli arrivi effettivi (4307) risultano *outlier* rispetto a quelli teorici (4391.98056).
- ▶ Effettuando la simulazione per 256 ripetizioni, i valori medi risultano:

| Arrivi | Dropped | Completamenti |
|------------|-----------|---------------|
| 4392.05859 | 218.32812 | 4173.73047 |

```
for 4094 jobs the service node statistics:  
# arrivals ..... = 4307  
# dropped ..... = 213  
# completions ..... = 4094  
avg interarrivals .. = 0.83595  
avg # in node ..... = 26.31702  
avg # in queue ..... = 3.69777  
avg utilization .... = 0.86997  
avg wait ..... = 23.21443  
avg delay ..... = 3.26183  
avg service ..... = 19.95260
```

* la simulazione riportata fa riferimento alla fascia oraria 14:00 - 15:00 ($\lambda_2 = 1.273056$)

VERIFICA

Doc & Pat D. (SSQ - M/M/1)

- ▶ $E(N_S) = E(N_Q) + \rho =$
 $= 0.01035 + 0.14898 = 0.15932$
- ▶ $E(T_S) = E(T_Q) + E(S) =$
 $= 1.29268 + 18.61538 = 19.90806$
- ▶ Arrivi = Completamenti + Dropped =
 $= 27 + 2 = 29$
- ▶ #expected_arrivals = 44.36344 **
** il calcolo è stato effettuato con lo script 'population.py'

```
for 27 jobs the service node statistics:  
# arrivals ..... = 29  
# dropped ..... = 2  
# completions ..... = 27  
avg interarrivals .. = 123.75219  
avg # in node ..... = 0.15932  
avg # in queue ..... = 0.01035  
avg utilization .... = 0.14898  
avg wait ..... = 19.90806  
avg delay ..... = 1.29268  
avg service ..... = 18.61538
```

VERIFICA

Doc & Pat D. (SSQ - M/M/1)

- ▶ Gli arrivi effettivi (29) risultano *outlier* rispetto a quelli teorici (44.36344).
- ▶ Effettuando la simulazione per 256 ripetizioni, i valori medi risultano:

| Arrivi | Dropped | Completamenti |
|----------|---------|---------------|
| 44.55469 | 2.28516 | 42.26953 |

```
for 27 jobs the service node statistics:  
# arrivals ..... = 29  
# dropped ..... = 2  
# completions ..... = 27  
avg interarrivals .. = 123.75219  
avg # in node ..... = 0.15932  
avg # in queue ..... = 0.01035  
avg utilization .... = 0.14898  
avg wait ..... = 19.90806  
avg delay ..... = 1.29268  
avg service ..... = 18.61538
```

VERIFICA

Police Control (MS - M/M/13/13)

- ▶ $E(N_S) = m\rho =$
 $= 13 \cdot 0.93494 = 12.15428$
- ▶ $E(T_S) = E(S) =$
 $= 39.90212 = 39.90212$
- ▶ Arrivi = Completamenti + Dropped =
 $= 1114 + 57 = 1171$
- ▶ **#lower_bound_arrivals = 1051.03543 ****
** il calcolo è stato effettuato con lo script 'population.py'

```
for 1114 jobs the service node statistics:  
# arrivals ..... = 1171  
# dropped ..... = 57  
# completions ..... = 1114  
avg interarrivals .. = 3.24444  
avg # in node ..... = 12.15428  
avg # in queue ..... = 0.00000  
avg utilization .... = 0.93494  
avg wait ..... = 39.90212  
avg delay ..... = 0.00000  
avg service ..... = 39.90212
```

VERIFICA

Police Control (MS - M/M/13/13)

- ▶ Gli arrivi effettivi (1171) rispettano il *lower bound* prefissato dal QoS (1051.03543, 25% degli arrivi).
- ▶ Il QoS prefissato è di avere un controllo da parte della polizia almeno del 25%:

$$\frac{\text{arrivi}_{PC}}{\text{arrivi}_T} = \frac{1171}{4078} = 0.2871 \quad (28.7\%)$$



POLICE CONTROL

```
# arrivals ..... = 1171  
# dropped ..... = 57  
# completions .... = 1114
```

TURNSTILES

```
# arrivals ..... = 4078  
# dropped ..... = 7  
# completions .... = 4071
```

* la simulazione riportata fa riferimento alla fascia oraria 14:00 - 15:00 ($\lambda_2 = 1.273056$)

VERIFICA

Turnstiles (MSQ - M/M/19)

- ▶ $E(N_S) = E(N_Q) + m\rho =$
 $= 3.04852 + 19 \cdot 0.8246 = 18.71583$
- ▶ $E(T_S) = E(T_Q) + E(S) =$
 $= 2.88084 + 14.80559 = 17.68643$
- ▶ Arrivi = Completamenti + Dropped =
 $= 4071 + 7 = 4078$
- ▶ #expected_arrivals = 4151.58995 **
** il calcolo è stato effettuato con lo script 'population.py'

```
for 4071 jobs the service node statistics:  
# arrivals ..... = 4078  
# dropped ..... = 7  
# completions ..... = 4071  
avg interarrivals .. = 0.94271  
avg # in node ..... = 18.71583  
avg # in queue ..... = 3.04852  
avg utilization .... = 0.82460  
avg wait ..... = 17.68643  
avg delay ..... = 2.88084  
avg service ..... = 14.80559
```

VERIFICA

Turnstiles (MSQ - M/M/19)

- ▶ Gli arrivi effettivi (4078) risultano *outlier* rispetto a quelli teorici (4151.58995).
- ▶ Effettuando la simulazione per 256 ripetizioni, i valori medi risultano:

| Arrivi | Dropped | Completamenti |
|------------|----------|---------------|
| 4147.52343 | 10.34375 | 4137.17969 |

```
for 4071 jobs the service node statistics:  
# arrivals ..... = 4078  
# dropped ..... = 7  
# completions ..... = 4071  
avg interarrivals .. = 0.94271  
avg # in node ..... = 18.71583  
avg # in queue ..... = 3.04852  
avg utilization .... = 0.82460  
avg wait ..... = 17.68643  
avg delay ..... = 2.88084  
avg service ..... = 14.80559
```

* la simulazione riportata fa riferimento alla fascia oraria 14:00 - 15:00 ($\lambda_2 = 1.273056$)

VERIFICA

Turnstiles D. (SSQ - M/M/1)

- ▶ $E(N_S) = E(N_Q) + \rho =$
 $= 0.09024 + 0.33620 = 0.42644$
- ▶ $E(T_S) = E(T_Q) + E(S) =$
 $= 3.42142 + 12.74700 = 16.16843$
- ▶ Arrivi = Completamenti + Dropped =
 $= 99 + 1 = 100$
- ▶ #expected_arrivals = 101.12848 **
** il calcolo è stato effettuato con lo script 'population.py'

```
for 99 jobs the service node statistics:  
# arrivals ..... = 100  
# dropped ..... = 1  
# completions ..... = 99  
avg interarrivals .. = 37.88205  
avg # in node ..... = 0.42644  
avg # in queue ..... = 0.09024  
avg utilization .... = 0.33620  
avg wait ..... = 16.16843  
avg delay ..... = 3.42142  
avg service ..... = 12.74700
```

* la simulazione riportata fa riferimento alla fascia oraria 14:00 - 15:00 ($\lambda_2 = 1.273056$)

VERIFICA

Turnstiles D. (SSQ - M/M/1)

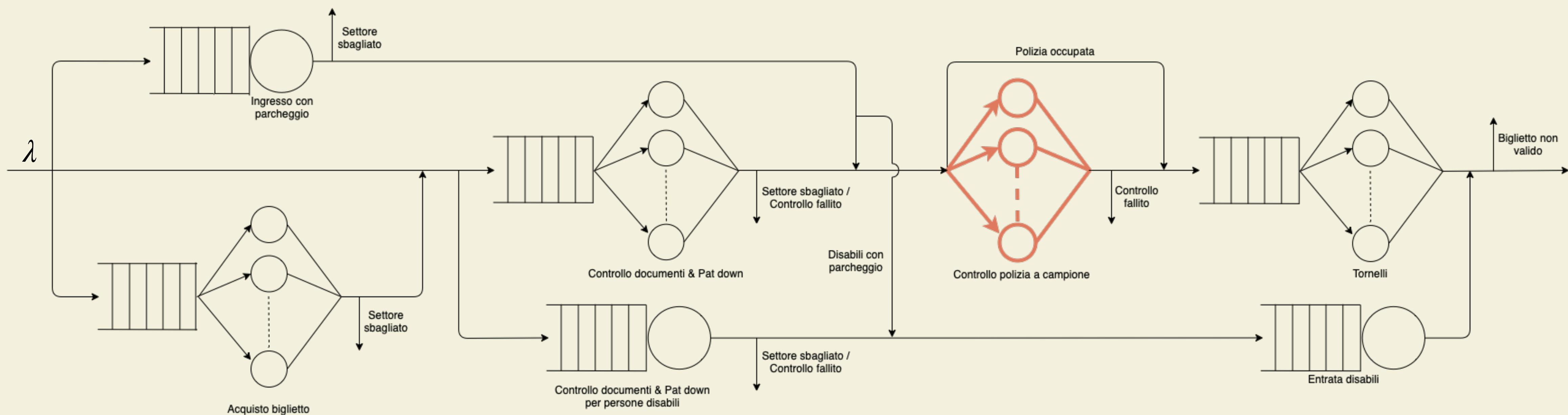
- ▶ Gli arrivi effettivi (100) risultano *outlier* rispetto a quelli teorici (101.12848).
- ▶ Effettuando la simulazione per 256 ripetizioni, i valori medi risultano:

| Arrivi | Dropped | Completamenti |
|-----------|---------|---------------|
| 101.69141 | 0.21094 | 101.48047 |

```
for 99 jobs the service node statistics:  
# arrivals ..... = 100  
# dropped ..... = 1  
# completions ..... = 99  
avg interarrivals .. = 37.88205  
avg # in node ..... = 0.42644  
avg # in queue ..... = 0.09024  
avg utilization .... = 0.33620  
avg wait ..... = 16.16843  
avg delay ..... = 3.42142  
avg service ..... = 12.74700
```

VALIDAZIONE

Ciò che ci si aspetta dal sistema è che, aumentando il numero di serventi nel centro ‘*Doc & Pat*’, la percentuale relativa al **QoS** del controllo della polizia (**25%**) diminuisca:



VALIDAZIONE

Ciò che ci si aspetta dal sistema è che, aumentando il numero di serventi nel centro ‘Doc & Pat’, la percentuale relativa al **QoS** del controllo della polizia (**25%**) diminuisca:

Serventi_DP = **22** Serventi_PC = **11**

POLICE CONTROL

arrivals = 1057
dropped = 44
completions = 1013

$$\frac{arrivi_{PC}}{arrivi_T} = \frac{1057}{4152} = 0,255$$

TURNSTILES

arrivals = 4152
dropped = 7
completions = 4145

Serventi_DP = **30** Serventi_PC = **11**

POLICE CONTROL

arrivals = 1019
dropped = 48
completions = 971

$$\frac{arrivi_{PC}}{arrivi_T} = \frac{1019}{4120} = 0,247$$

TURNSTILES

arrivals = 4120
dropped = 10
completions = 4110

VALIDAZIONE

Sono stati confrontati i risultati delle simulazioni con i valori teorici scaturiti dalle seguenti formule:

$$E(T_S)_{KP} = \frac{\rho E(S)}{1 - \rho} + E(S)$$

$$p(0) = \left[\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1-\rho)} \right]^{-1}$$

$$\lambda' = \lambda(1 - p_{loss})$$
$$\rho = \frac{\lambda'}{\mu}$$

$$E(T_S)_{Erlang} = \frac{P_Q E(S)}{1 - \rho} + E(S_i)$$

$$P_Q = \frac{(m\rho)^m}{m!(1-\rho)} p(0)$$

$$* \pi_m = \frac{\left(\frac{\lambda}{\mu}\right)^m / m!}{\sum_{j=0}^m \left(\frac{\lambda}{\mu}\right)^j \frac{1}{j!}}$$

$$E(T_S) = \sum_{i=0}^m v_i \cdot E(T_{S_i})$$

$$E(T_Q) = \frac{P_Q E(S)}{1 - \rho}$$

VALIDAZIONE

Configurazioni

Le diverse **configurazioni di serventi** utilizzate per le **256 simulazioni** della validazione, sono quelle risultate ottimali per rispettare gli obiettivi e i QoS:

| | Park Entry | Ticket Buy | Doc & Pat | Doc & Pat D. | Police Control | Turnstiles | Turnstiles D. |
|---------------------|------------|------------|-----------|--------------|----------------|------------|---------------|
| 13:00 - 14:00 | 1 | 5 | 13 | 1 | 6 | 9 | 1 |
| 14:00 - 15:00 | 1 | 11 | 26 | 1 | 13 | 19 | 1 |
| 15:00 - 16:00 | 1 | 2 | 3 | 1 | 2 | 2 | 1 |

* il calcolo è stato effettuato
con lo script ‘queue_time.py’

FASCIA 13:00 – 14:00

| Statistica | Risultato Analitico * | Risultato Sperimentale ($\alpha = 0.05$) |
|------------------------------|-----------------------|---|
| $E(T_{Q_{PARK_ENTRY}})$ | 16.33205 | 15.97357 ± 0.77346 |
| $E(T_{Q_{TICKET_BUY}})$ | 75.52453 | 77.72653 ± 6.08385 |
| $E(T_{Q_{DOC_PAT}})$ | 6.00957 | 5.96558 ± 0.19859 |
| $E(T_{Q_{DOC_PAT_D}})$ | 2.56674 | 2.38105 ± 0.19344 |
| $E(T_{Q_{POLICE_CONTROL}})$ | 0.00000 | 0.00000 ± 0.00000 |
| $E(T_{Q_{TURNSTILES}})$ | 9.55838 | 9.67299 ± 0.37824 |
| $E(T_{Q_{TURNSTILES_D}})$ | 3.62097 | 3.40926 ± 0.22181 |

* il calcolo è stato effettuato
con lo script ‘queue_time.py’

FASCIA 14:00 – 15:00

| Statistica | Risultato Analitico * | Risultato Sperimentale ($\alpha = 0.05$) |
|------------------------------|------------------------------|--|
| $E(T_{Q_{PARK_ENTRY}})$ | 97.02908 | 92.91735 ± 7.42203 |
| $E(T_{Q_{TICKET_BUY}})$ | 22.94744 | 22.78496 ± 2.04092 |
| $E(T_{Q_{DOC_PAT}})$ | 8.33916 | 8.11687 ± 0.43658 |
| $E(T_{Q_{DOC_PAT_D}})$ | 6.54152 | 6.16607 ± 0.38180 |
| $E(T_{Q_{POLICE_CONTROL}})$ | 0.00000 | 0.00000 ± 0.00000 |
| $E(T_{Q_{TURNSTILES}})$ | 5.2815 | 5.26622 ± 0.27694 |
| $E(T_{Q_{TURNSTILES_D}})$ | 10.92325 | 10.59737 ± 0.49544 |

* il calcolo è stato effettuato
con lo script ‘queue_time.py’

FASCIA 15:00 – 16:00

| Statistica | Risultato Analitico * | Risultato Sperimentale ($\alpha = 0.05$) |
|------------------------------|-----------------------|---|
| $E(T_{Q_{PARK_ENTRY}})$ | 1.86973 | 1.76706 ± 0.11201 |
| $E(T_{Q_{TICKET_BUY}})$ | 12.72913 | 12.81082 ± 0.40134 |
| $E(T_{Q_{DOC_PAT}})$ | 6.87433 | 6.88404 ± 0.08748 |
| $E(T_{Q_{DOC_PAT_D}})$ | 0.38591 | 0.36980 ± 0.05437 |
| $E(T_{Q_{POLICE_CONTROL}})$ | 0.00000 | 0.00000 ± 0.00000 |
| $E(T_{Q_{TURNSTILES}})$ | 11.84891 | 11.72581 ± 0.13766 |
| $E(T_{Q_{TURNSTILES_D}})$ | 0.50170 | 0.48458 ± 0.03605 |

INFINITE HORIZON SIMULATION

La simulazione ad **orizzonte infinito** è stata effettuata su un lasso di tempo molto più lungo rispetto al tempo reale.

Si considera una singola fascia oraria per volta.

Il sistema viene assunto **statico**:

- ▶ il **numero di serventi** rimane invariato
- ▶ il **flusso di arrivo** rimane invariato

Gli **obiettivi** principali sono:

- ▶ analizzare il comportamento del sistema allo stato stazionario
- ▶ individuare la configurazione migliore per rispettare i **QoS** e **minimizzare i costi**

INFINITE HORIZON SIMULATION

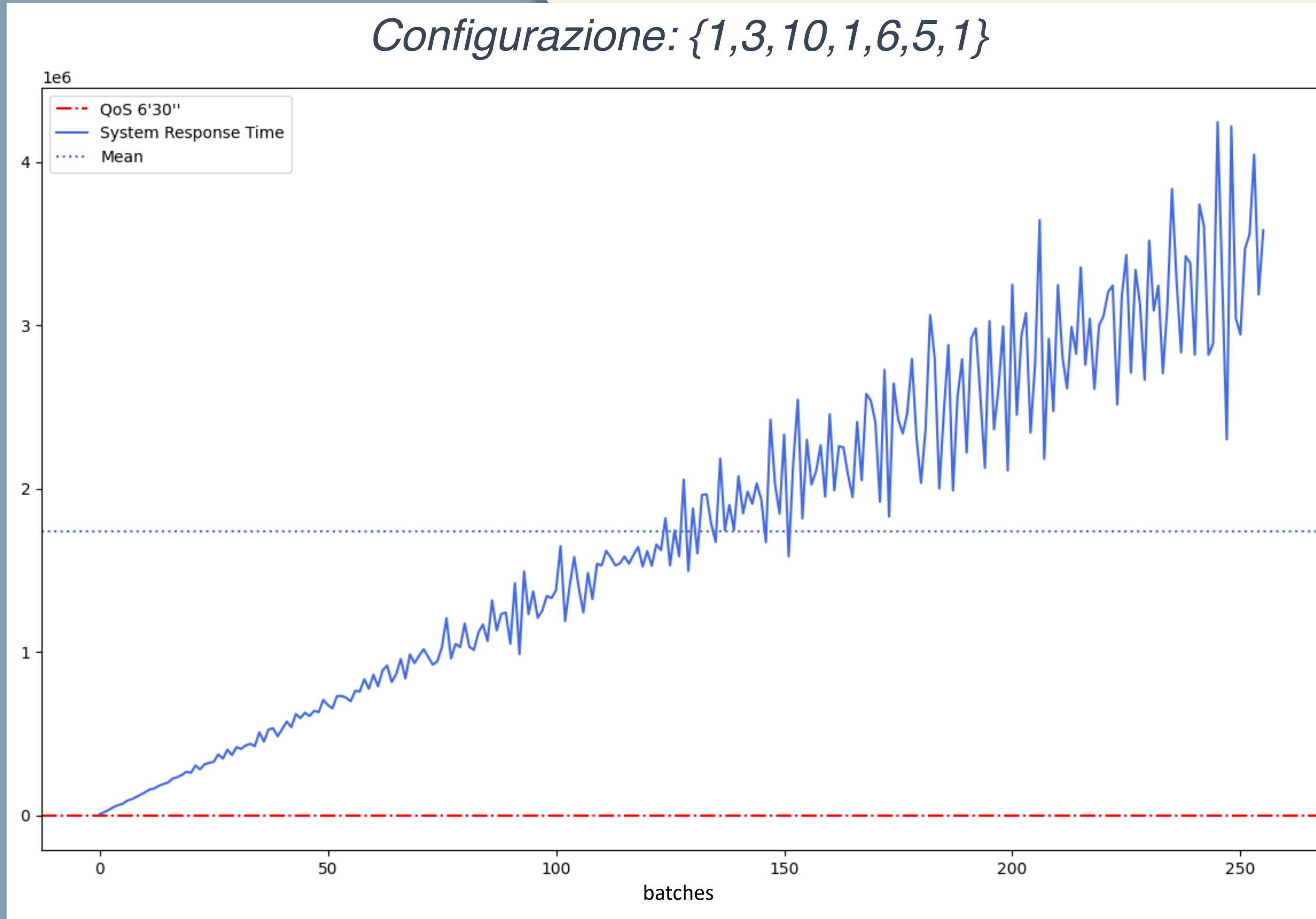
È stato utilizzato il metodo delle *Batch Means*:

- ▶ è stata suddivisa l'esecuzione in $k = 256$ batch di $b = 8192$ jobs
 - k e b scelti usando il metodo di Banks, Carson, Nelson, e Nicol
 - fissato k , si è cercato il valore di b per cui l'*autocorrelazione* del campione sia minore di 0.2 per lag $j=1$
- ▶ sono state calcolate le statistiche per ogni batch
- ▶ è stato generato un campione di k batches indipendenti su cui studiare la media campionaria

INFINITE HORIZON SIMULATION

1° Fascia 13:00 – 14:00

Configurazione: {1,3,10,1,6,5,1}

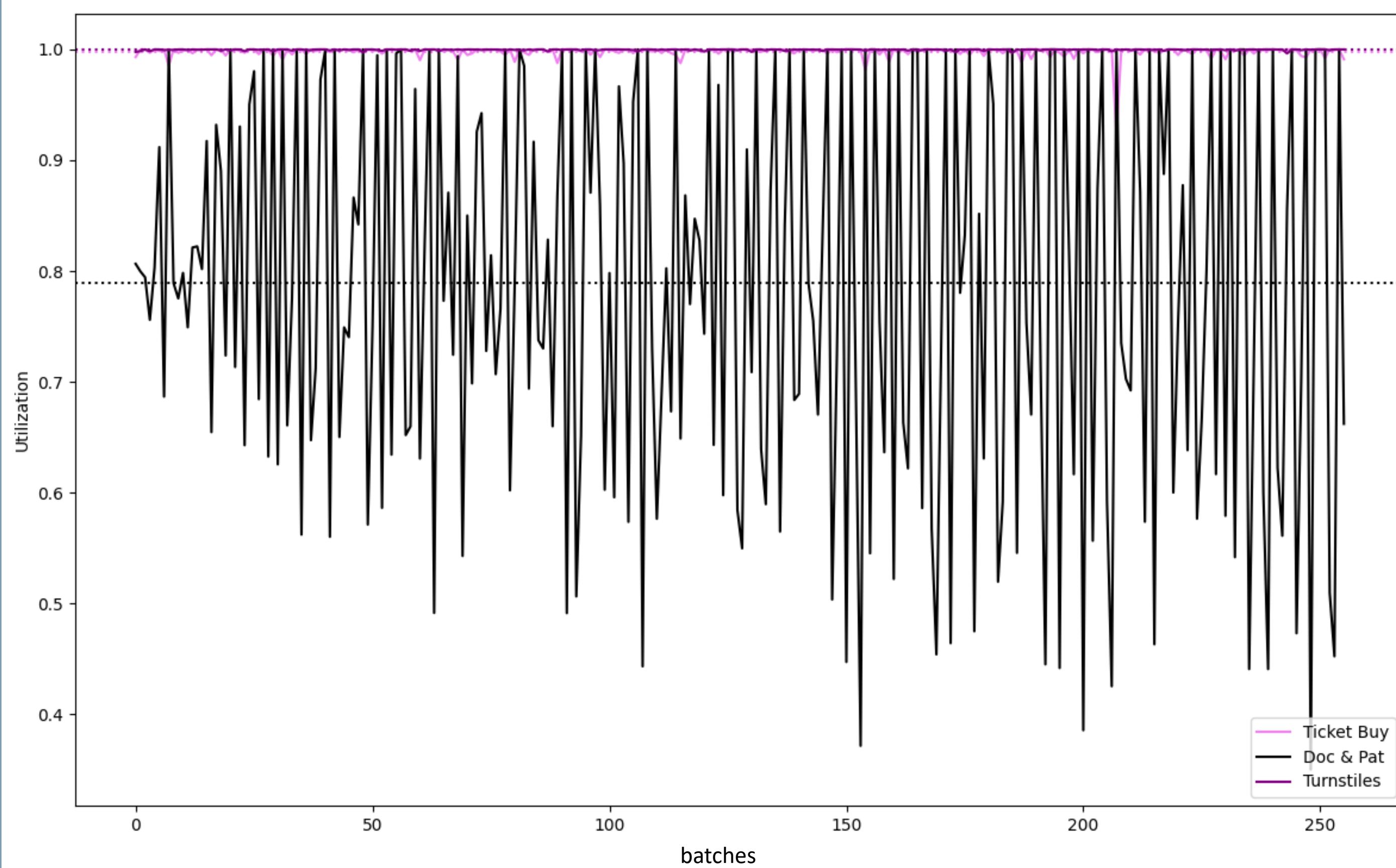


- ▶ Il sistema è fortemente sottodimensionato e non stazionario
- ▶ Il tempo di risposta del sistema cresce all'infinito

INFINITE HORIZON SIMULATION

1° Fascia 13:00 – 14:00

Configurazione: {1,3,10,1,6,5,1}



Analizzando le utilizzazioni:

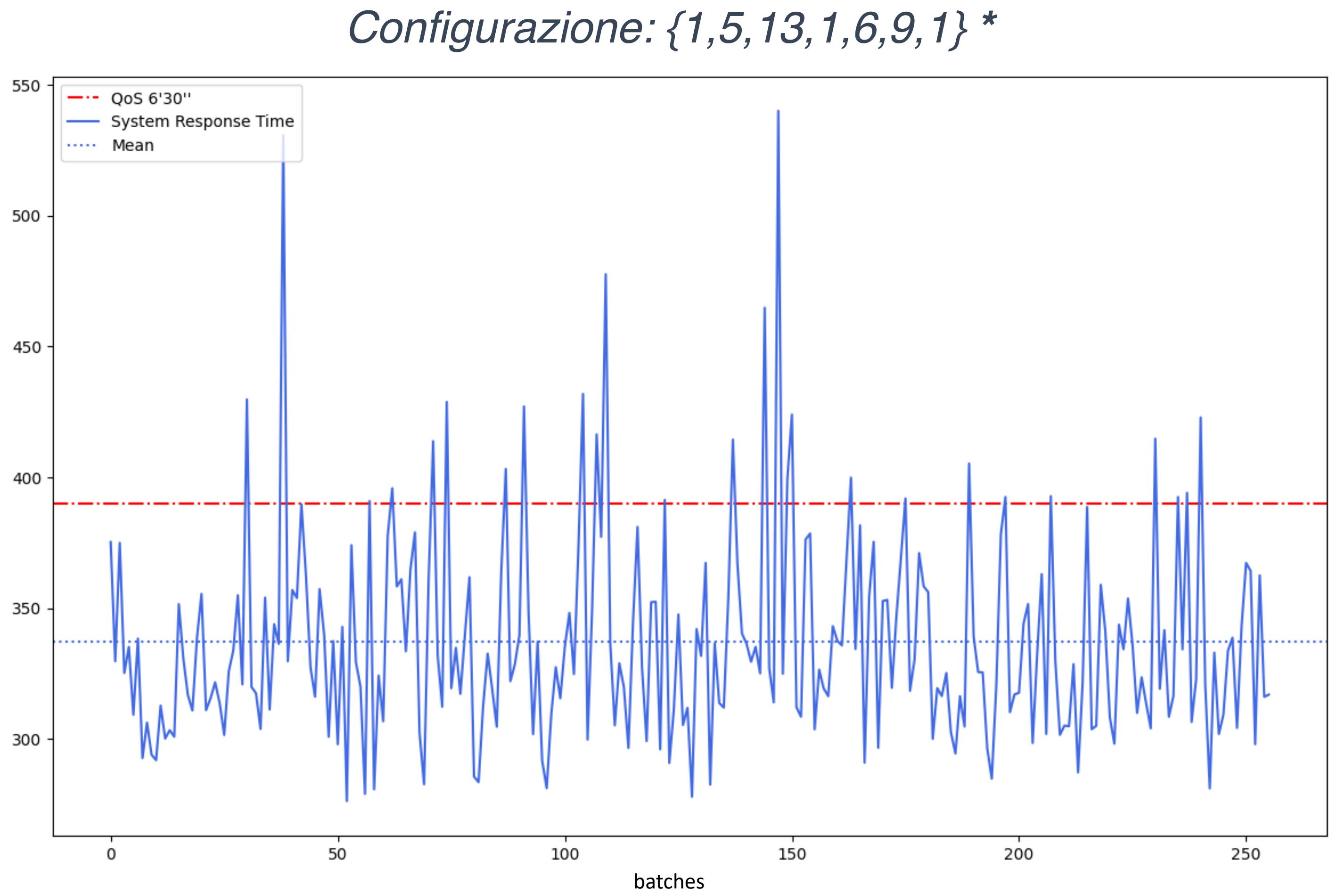
- ▶ **Ticket Buy** e **Turnstiles** hanno utilizzazioni pari a 1
- ▶ **Doc & Pat** ha media di utilizzazione inferiore a 0.8, tuttavia è molto variabile e spesso arriva a 1, questo perchè ha arrivi che dipendono anche da **Ticket Buy**

* Ridimensionamento prima fascia:
 $\{1,3,10,1,6,5,1\} \rightarrow \{1,5,13,1,6,9,1\}$

INFINITE HORIZON SIMULATION

1° Fascia 13:00 – 14:00

Configurazione: $\{1,5,13,1,6,9,1\}$ *



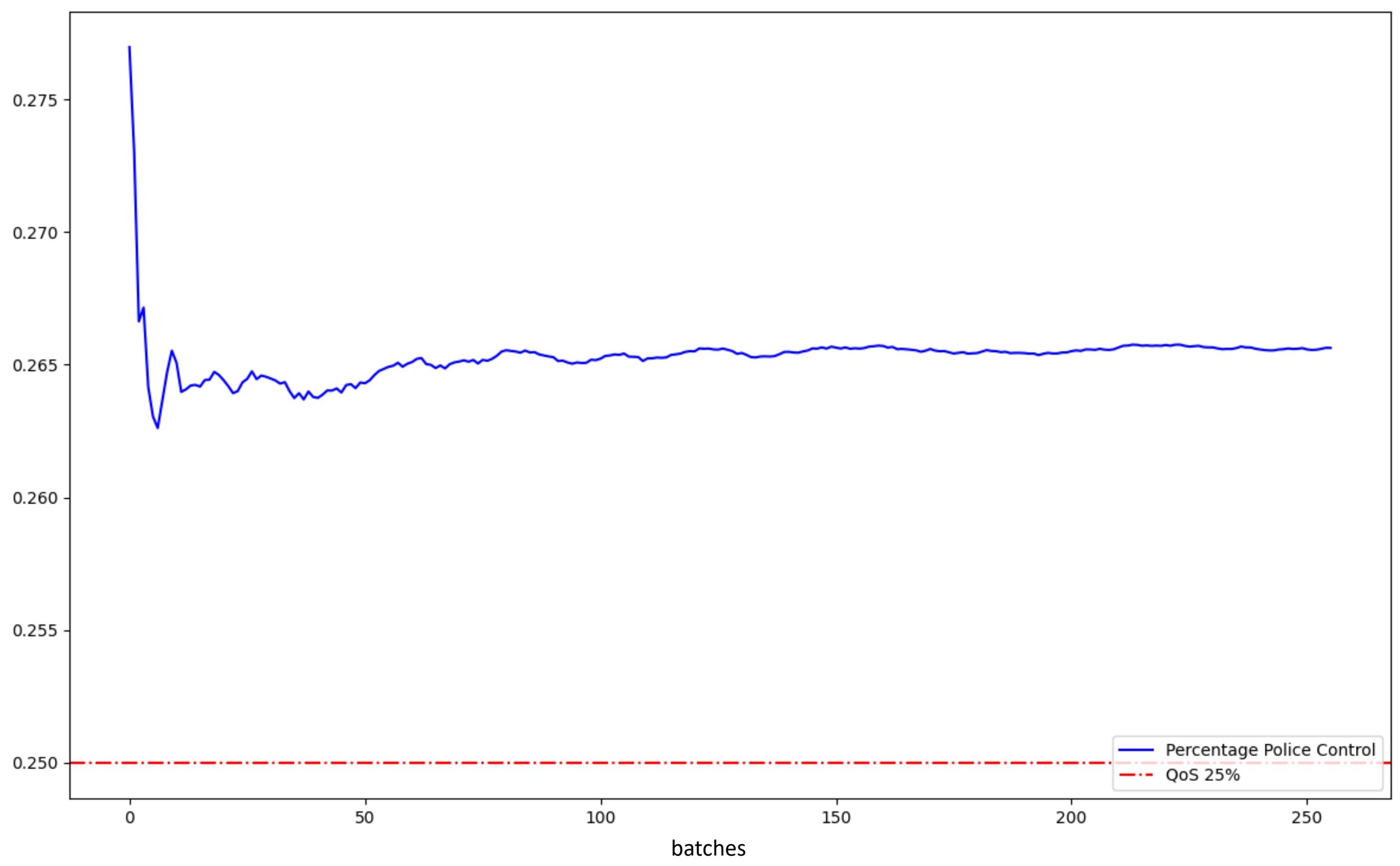
- ▶ Il sistema è **stazionario**
- ▶ Risultato Analitico (da '*queue_time.py*') : **343.61223**
- ▶ Risultato Sperimentale ($\alpha = 0.05$):
 337.78158 ± 5.526125
- ▶ Rispetta il QoS di 390 secondi (6 minuti e mezzo)

* Ridimensionamento prima fascia:
 $\{1,3,10,1,6,5,1\} \rightarrow \{1,5,13,1,6,9,1\}$

INFINITE HORIZON SIMULATION

1° Fascia 13:00 – 14:00

Configurazione: $\{1,5,13,1,6,9,1\}$ *



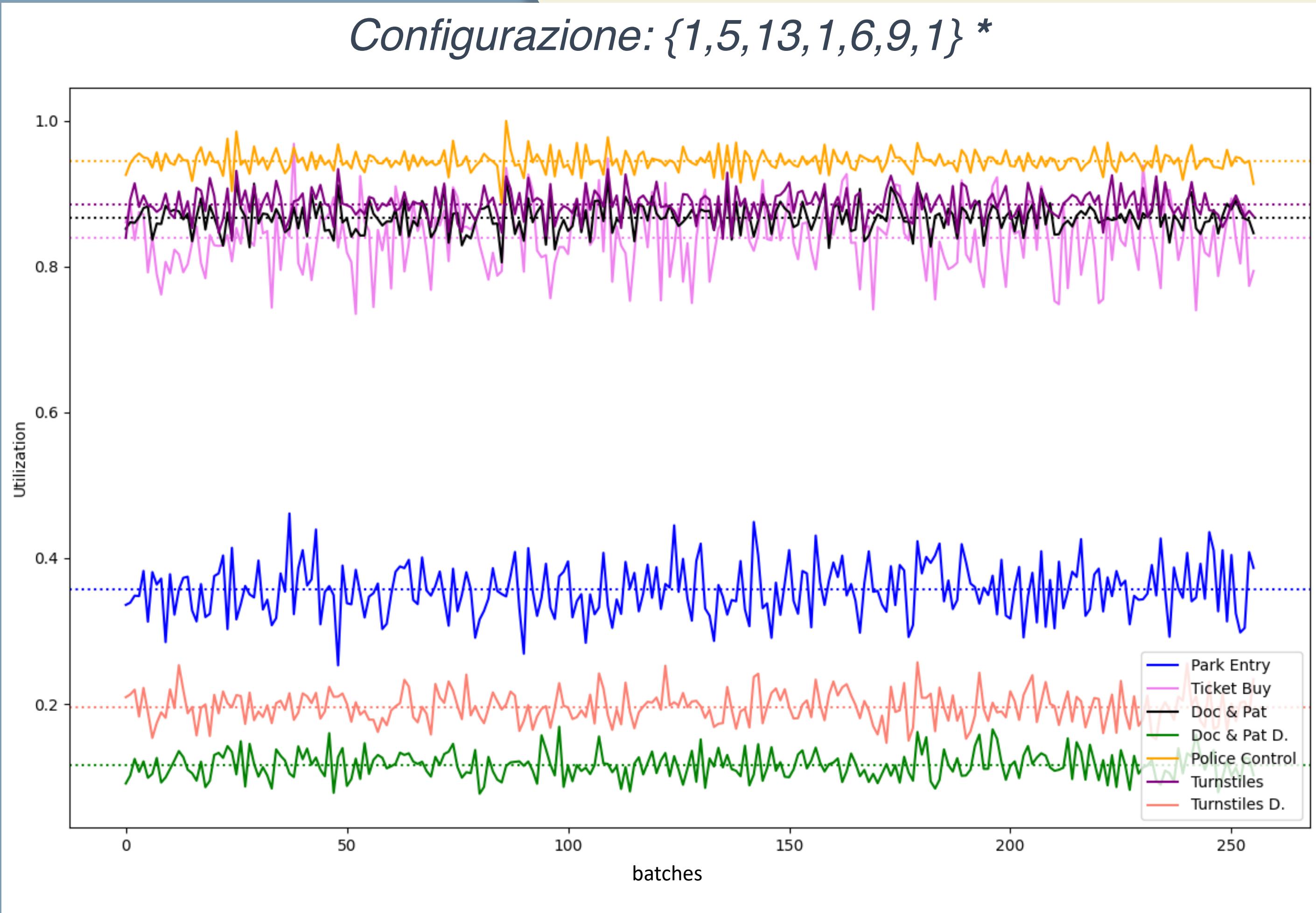
- ▶ Viene rispettato anche il **QoS** che richiede una percentuale minima di controllo del **25%** da parte della polizia
- ▶ La media è circa del **26.5%** ed è quindi il **minimo numero di poliziotti** per rispettare il **QoS**
- ▶ La **percentuale** inizialmente è molto alta poiché i primi utenti verranno controllati tutti

* Ridimensionamento prima fascia:
 $\{1,3,10,1,6,5,1\} \rightarrow \{1,5,13,1,6,9,1\}$

INFINITE HORIZON SIMULATION

1° Fascia 13:00 – 14:00

Configurazione: $\{1,5,13,1,6,9,1\}$ *



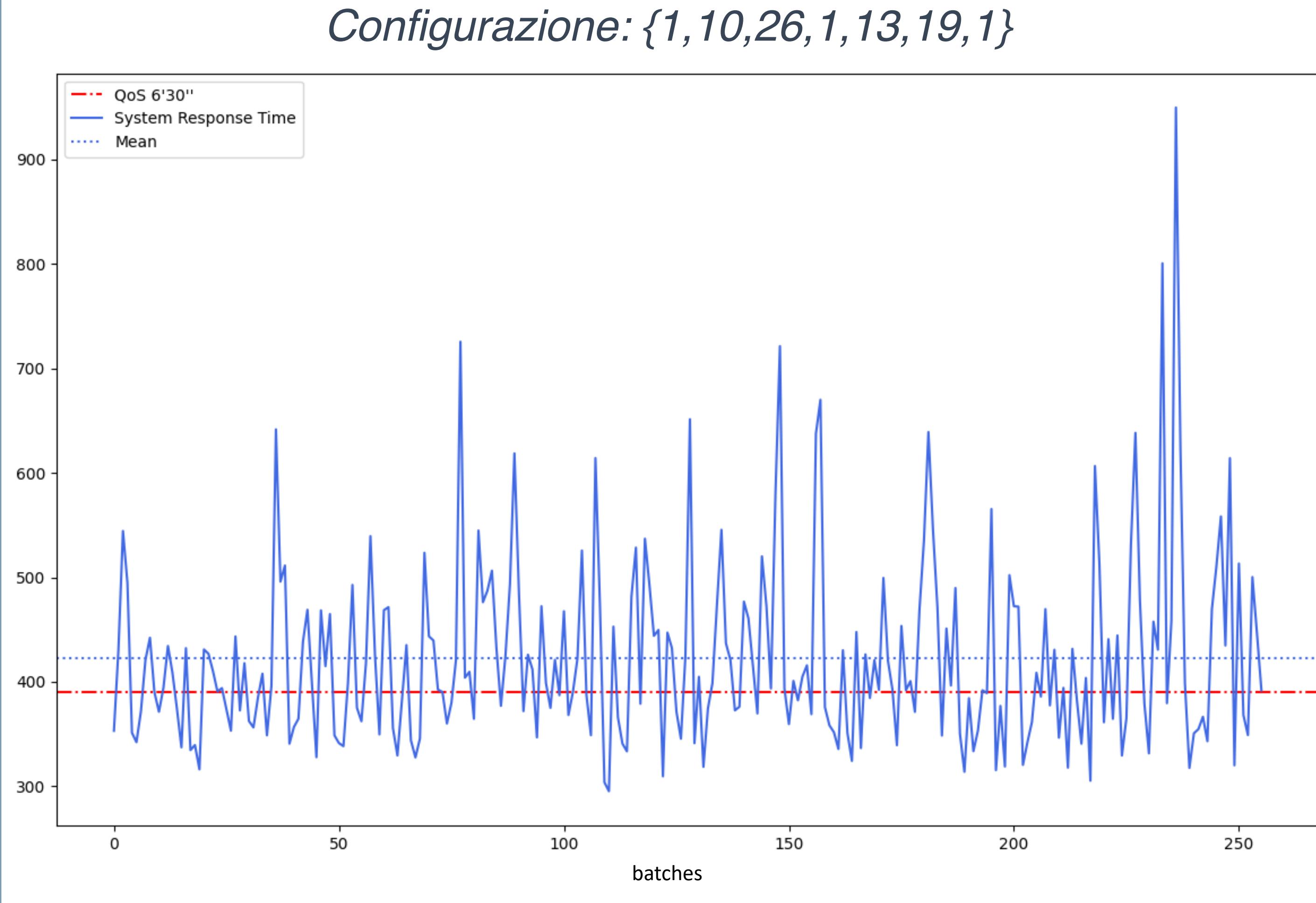
Analizzando le utilizzazioni:

- ▶ Police Control ha l'utilizzazione vicina a 1, non avendo coda significa che lavora nel modo corretto
- ▶ I multiserver hanno media di utilizzazione compresa tra 0.8 e 0.9, quindi $\{1,5,13,1,6,9,1\}$ è la configurazione minima per la stazionarietà

INFINITE HORIZON SIMULATION

2° Fascia 14:00 – 15:00

Configurazione: {1, 10, 26, 1, 13, 19, 1}

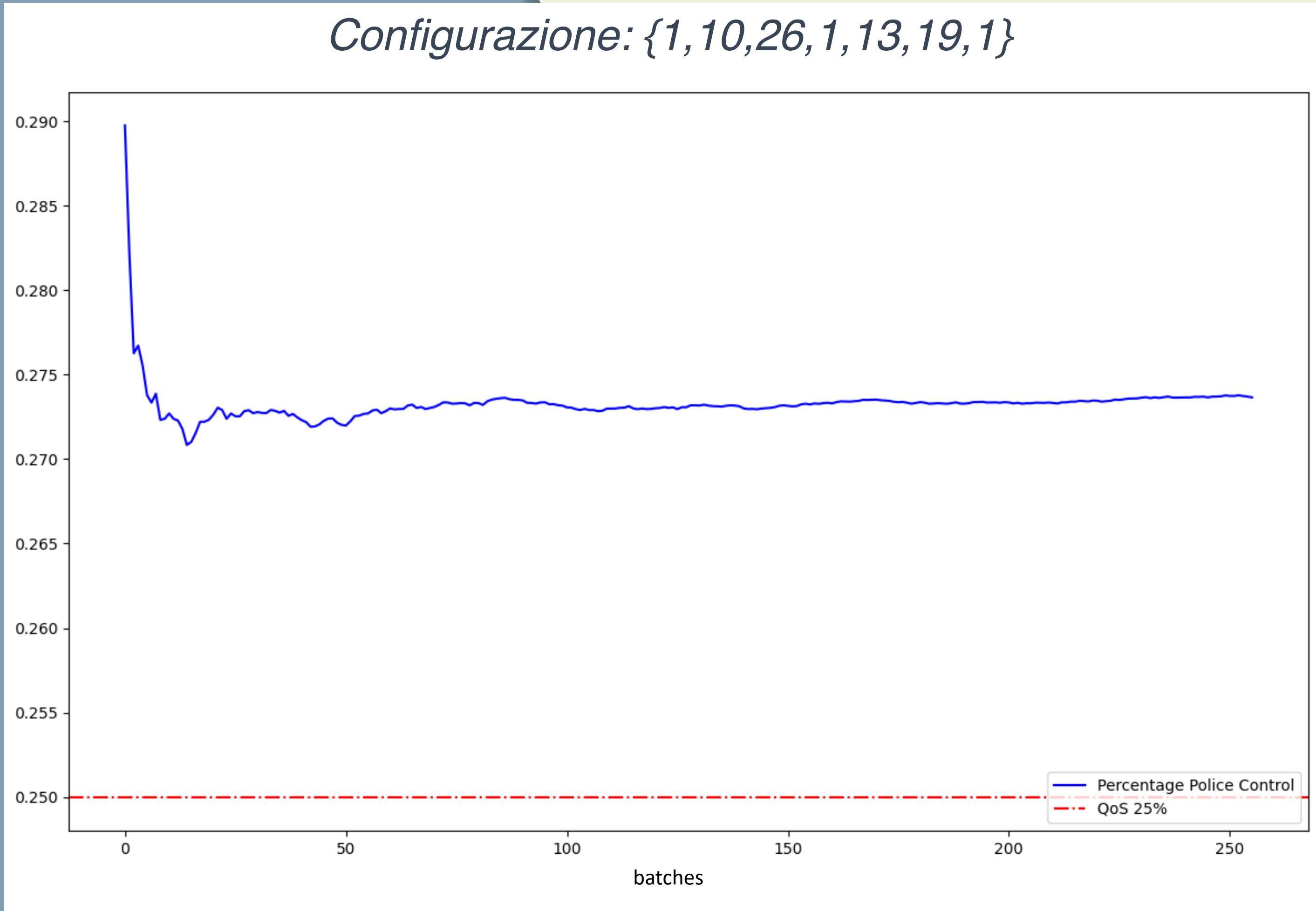


- ▶ Il sistema è **stazionario**
- ▶ Ma ha un **tempo di risposta** medio **superiore al QoS** (6 minuti e mezzo)

INFINITE HORIZON SIMULATION

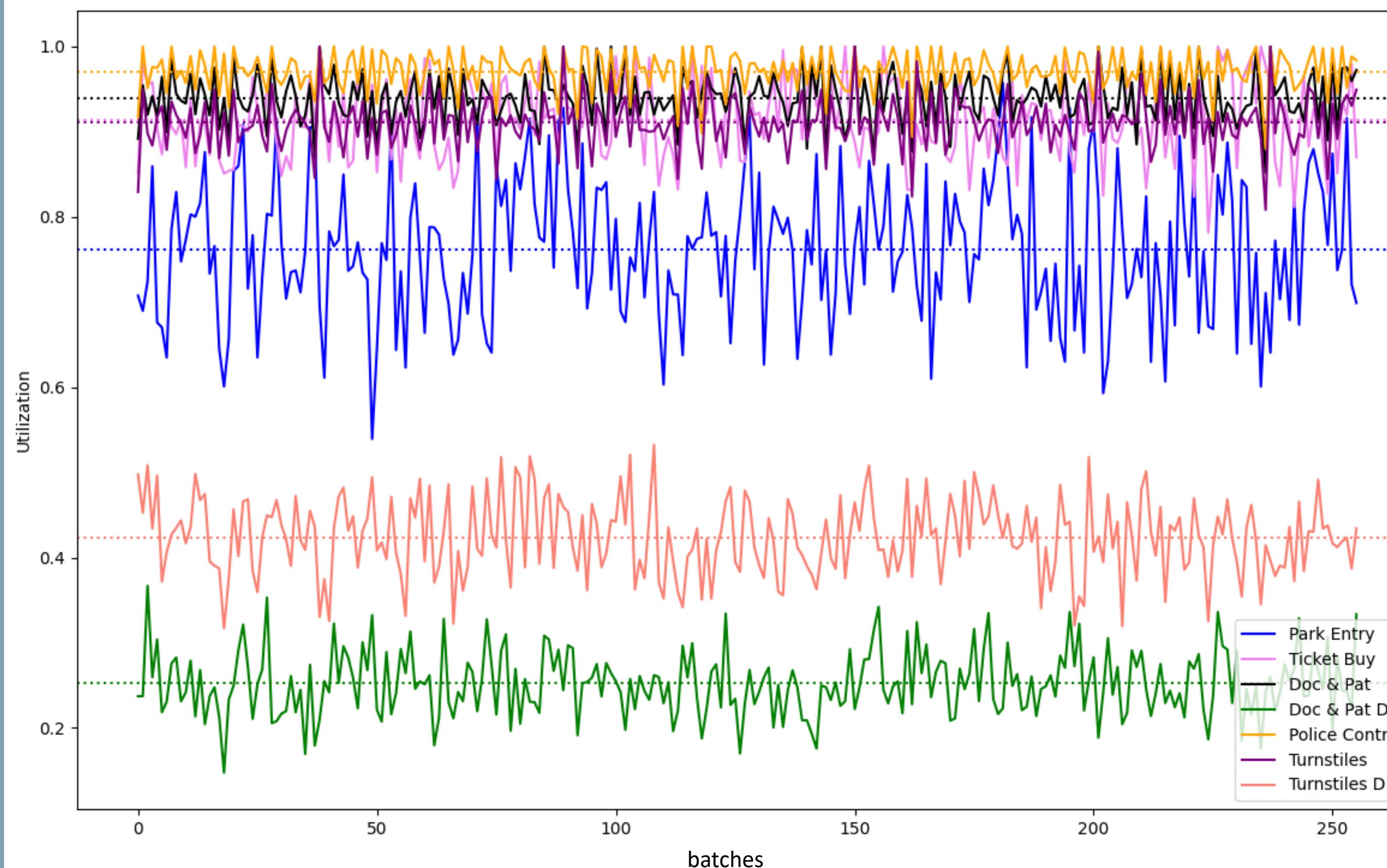
2° Fascia 14:00 – 15:00

Configurazione: {1,10,26,1,13,19,1}



- ▶ Il QoS che richiede una percentuale minima di controllo del **25%** da parte della polizia **viene rispettato**
- ▶ La media è compresa tra il **27%** e il **27.5%** ed è quindi il **minimo numero di poliziotti per rispettare il QoS**

Configurazione: {1,10,26,1,13,19,1}



Analizzando le utilizzazioni:

- I multiserver hanno media di utilizzazione vicina a 0.9, quindi **{1,10,26,1,13,19,1}** è la **configurazione minima** per la stazionarietà ma che **non rispetta il QoS** del tempo di risposta

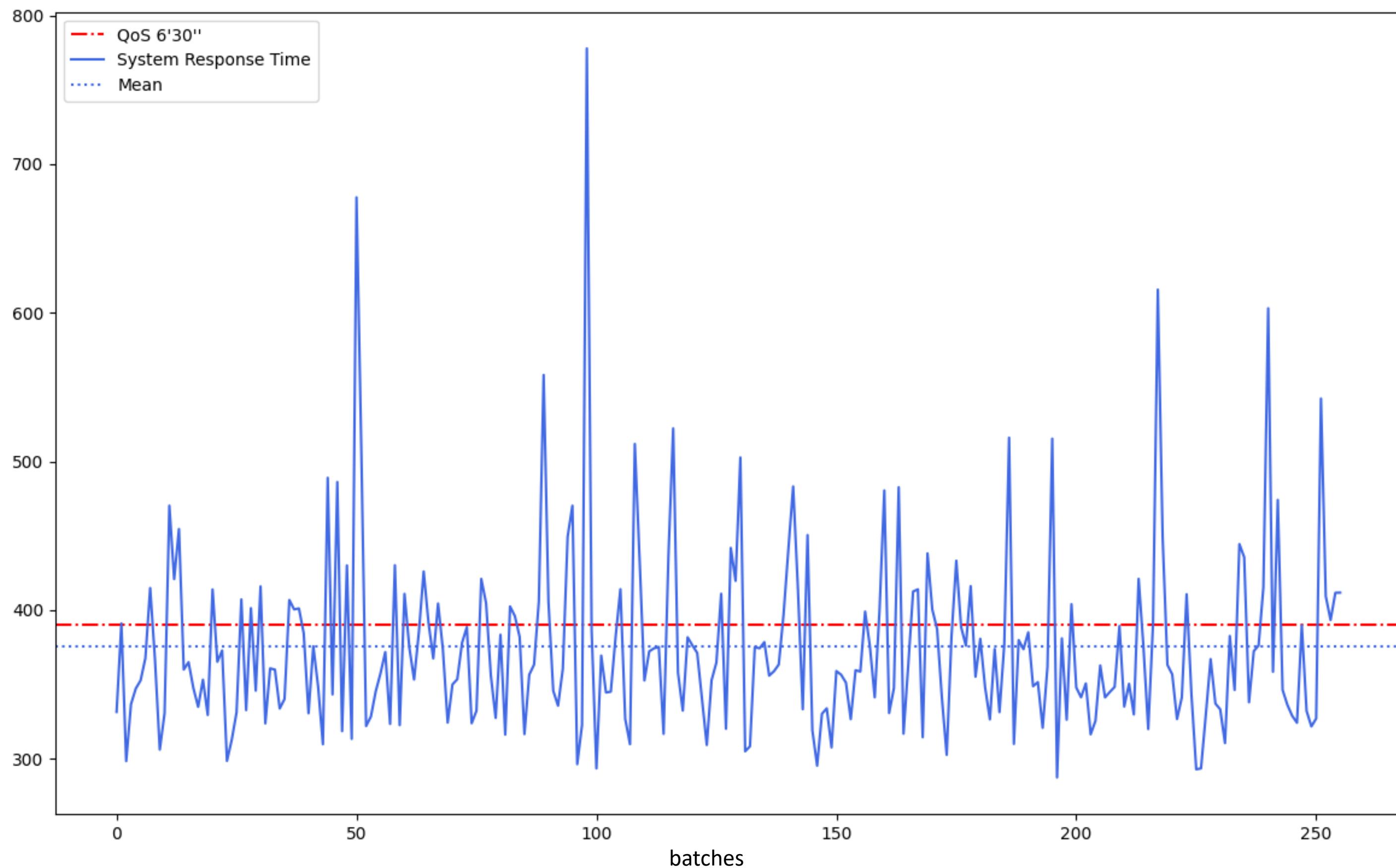
INFINITE HORIZON SIMULATION

2° Fascia 14:00 – 15:00

* Ridimensionamento seconda fascia:

$$\{1, 10, 26, 1, 13, 19, 1\} \rightarrow \{1, 11, 26, 1, 13, 19, 1\}$$

Configurazione: $\{1, 11, 26, 1, 13, 19, 1\}$ *



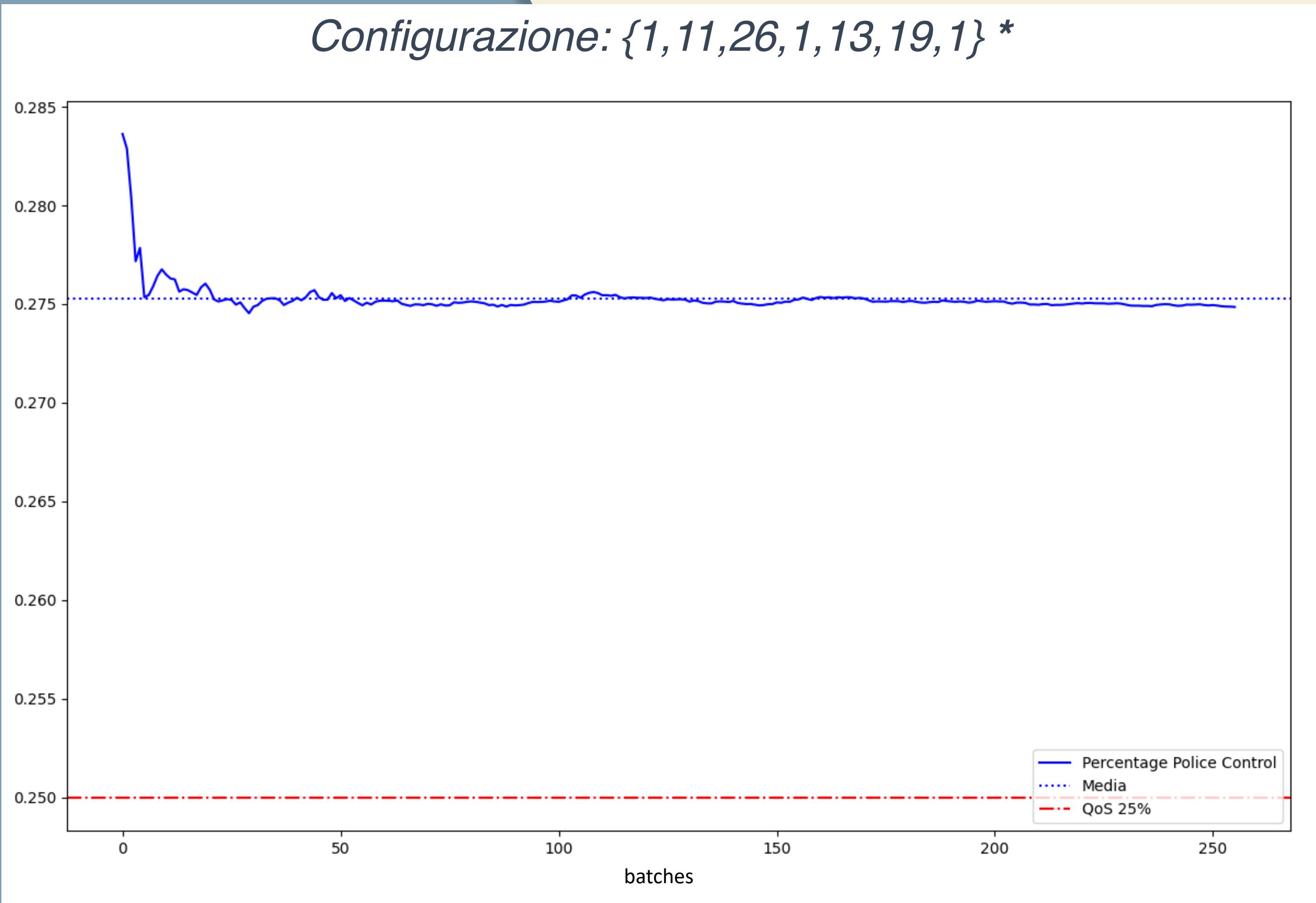
- ▶ Il sistema è **stazionario**
- ▶ Risultato Analitico (da 'queue_time.py'): **381.06194**
- ▶ Risultato Sperimentale ($\alpha = 0.05$):
 378.34666 ± 8.73926
- ▶ Rispetta il QoS di 390 secondi (6 minuti e mezzo)

* Ridimensionamento seconda fascia:
 $\{1, 10, 26, 1, 13, 19, 1\} \rightarrow \{1, 11, 26, 1, 13, 19, 1\}$

INFINITE HORIZON SIMULATION

2° Fascia 14:00 – 15:00

Configurazione: $\{1, 11, 26, 1, 13, 19, 1\}$ *



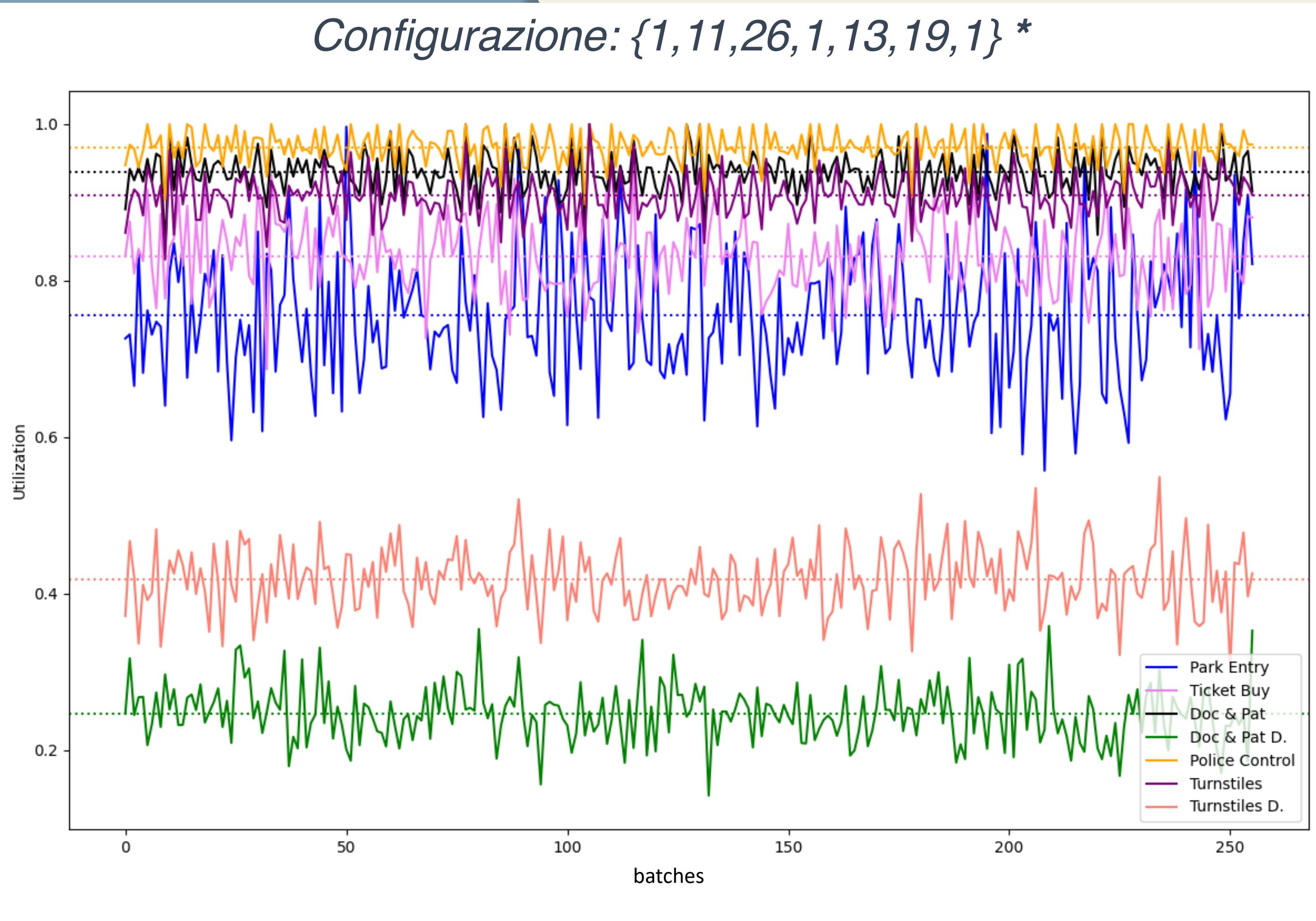
- ▶ Viene rispettato anche il **QoS** che richiede una percentuale minima di controllo del **25%** da parte della polizia
- ▶ La media è circa del **27.5%** ed è quindi il **minimo numero di poliziotti** per rispettare il **QoS**

* Ridimensionamento seconda fascia:
 $\{1, 10, 26, 1, 13, 19, 1\} \rightarrow \{1, 11, 26, 1, 13, 19, 1\}$

INFINITE HORIZON SIMULATION

2° Fascia 14:00 – 15:00

Configurazione: $\{1, 11, 26, 1, 13, 19, 1\}$ *



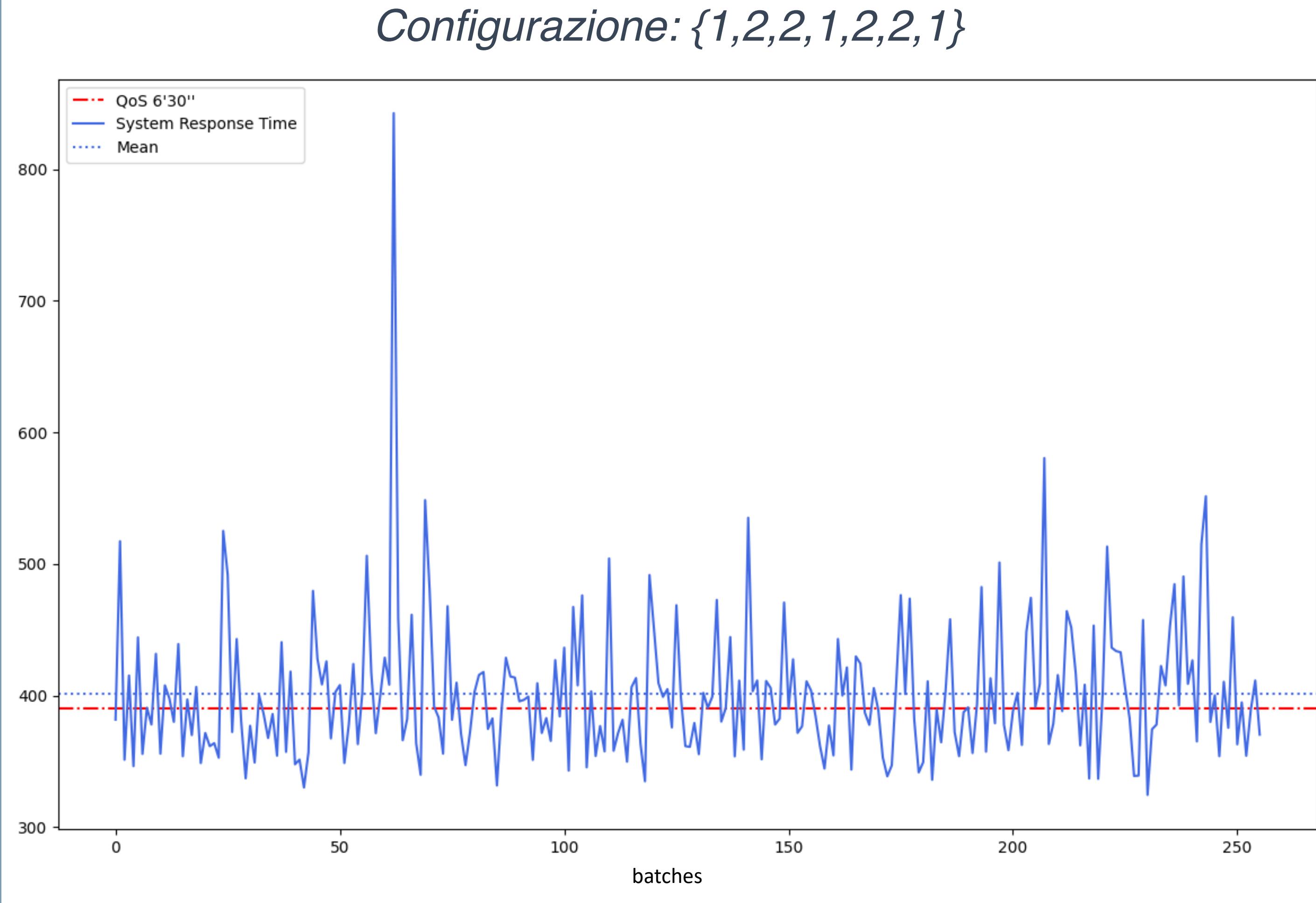
Analizzando le utilizzazioni:

- I multiserver hanno media di utilizzazione compresa tra 0.8 e 0.9, quindi $\{1, 11, 26, 1, 13, 19, 1\}$ è la configurazione minima per la stazionarietà e per rispettare il QoS del tempo di risposta

INFINITE HORIZON SIMULATION

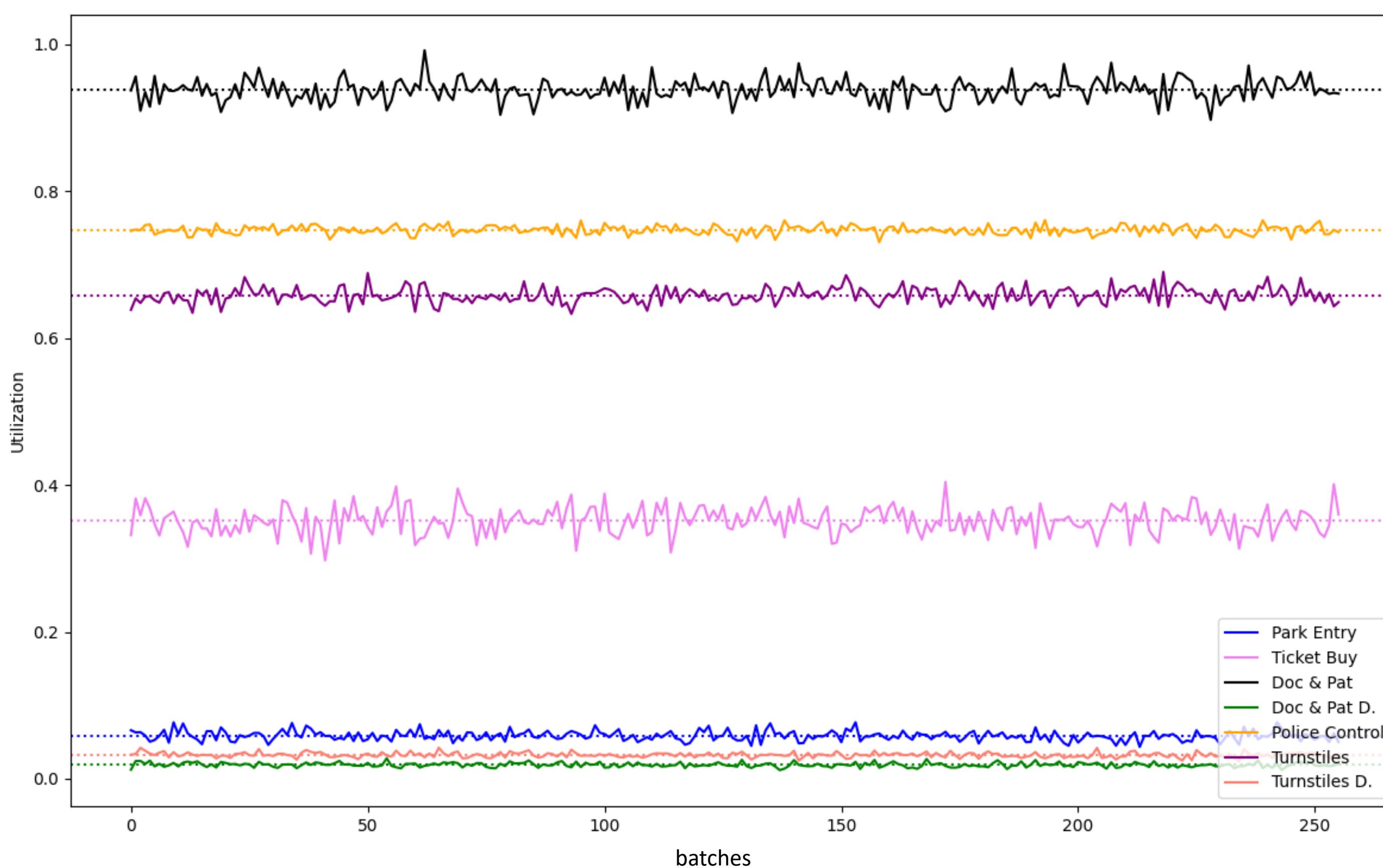
3° Fascia 15:00 – 16:00

Configurazione: {1,2,2,1,2,2,1}



- ▶ Il sistema è **stazionario**
- ▶ Ma ha un **tempo di risposta** medio **superiore al QoS** (6 minuti e mezzo)

Configurazione: {1,2,2,1,2,2,1}



Analizzando le utilizzazioni:

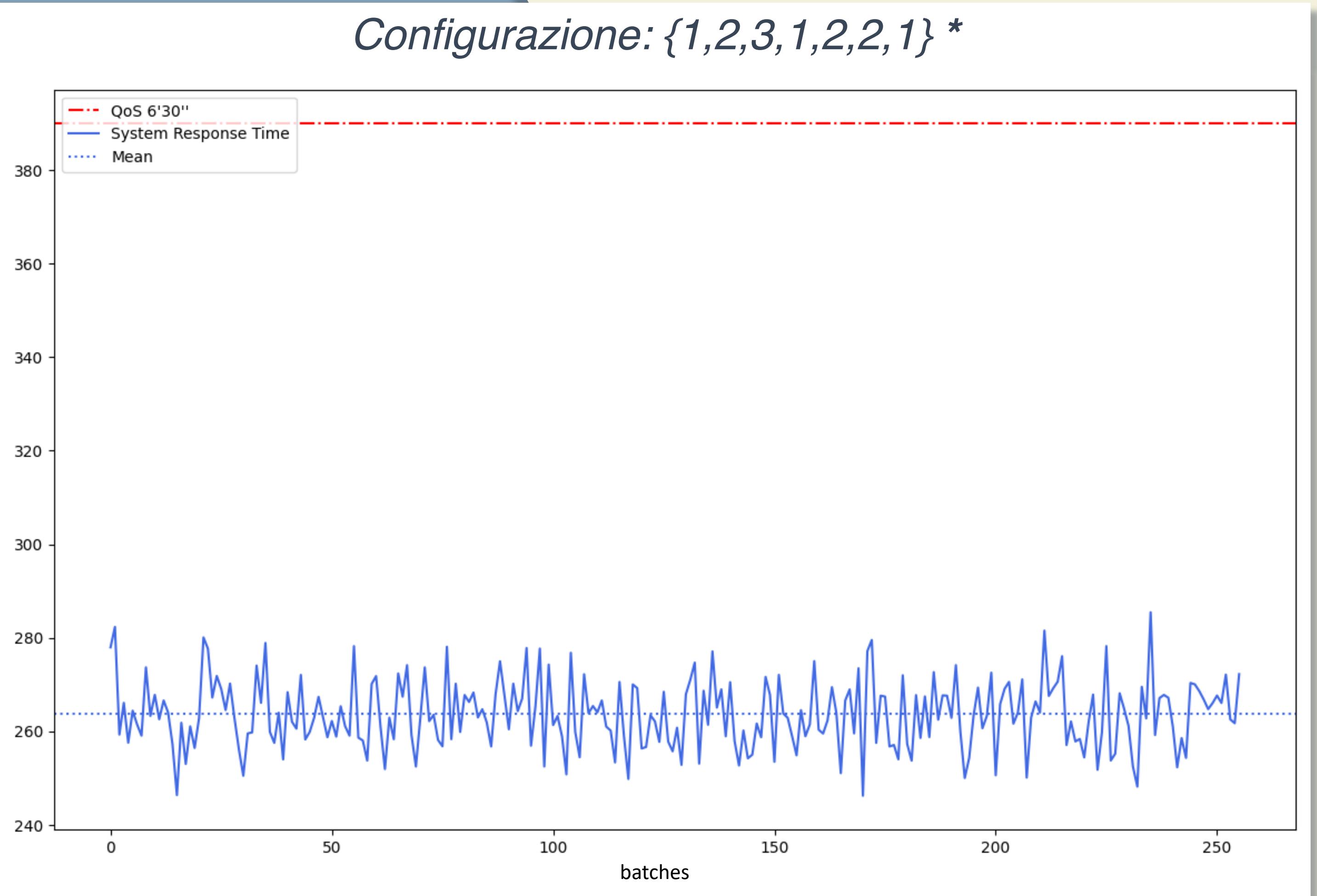
- ▶ Doc & Pat è il centro con l'utilizzazione più alta
- ▶ {1,2,2,1,2,2,1} è la configurazione minima per la stazionarietà e per non variare il modello (multi-server → single-server) ma che non rispetta il QoS del tempo di risposta

* Ridimensionamento terza fascia:
 $\{1,2,2,1,2,2,1\} \rightarrow \{1,2,3,1,2,2,1\}$

INFINITE HORIZON SIMULATION

3° Fascia 15:00 – 16:00

Configurazione: $\{1,2,3,1,2,2,1\}$ *



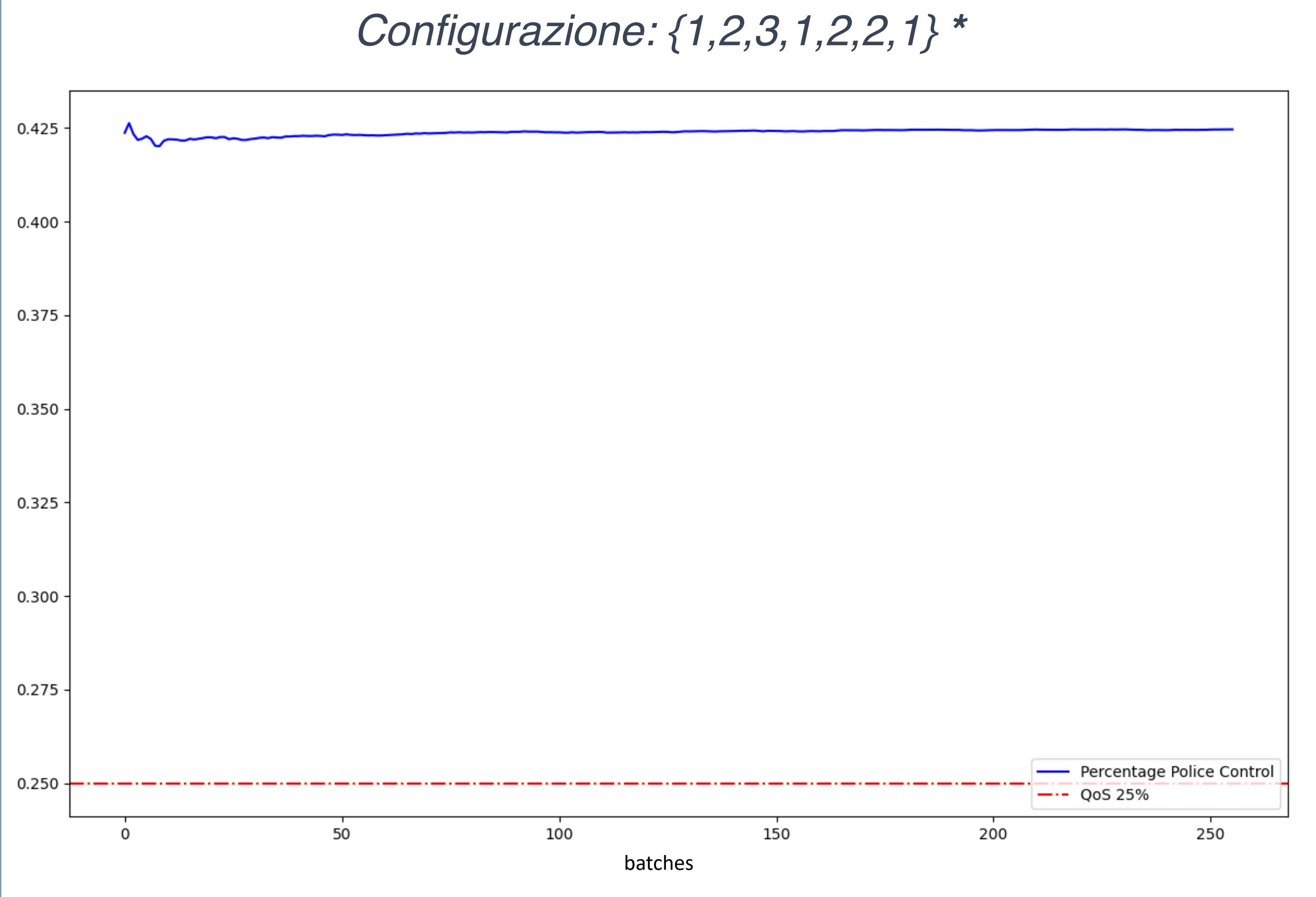
- ▶ Il sistema è **stazionario**
- ▶ Risultato Analitico (da '*queue_time.py*') : **264.20971**
- ▶ Risultato Sperimentale ($\alpha = 0.05$):
 263.83884 ± 0.93551
- ▶ Rispetta il QoS di 390 secondi (6 minuti e mezzo)

* Ridimensionamento terza fascia:
 $\{1,2,2,1,2,2,1\} \rightarrow \{1,2,3,1,2,2,1\}$

INFINITE HORIZON SIMULATION

3° Fascia 15:00 – 16:00

Configurazione: $\{1,2,3,1,2,2,1\}$ *



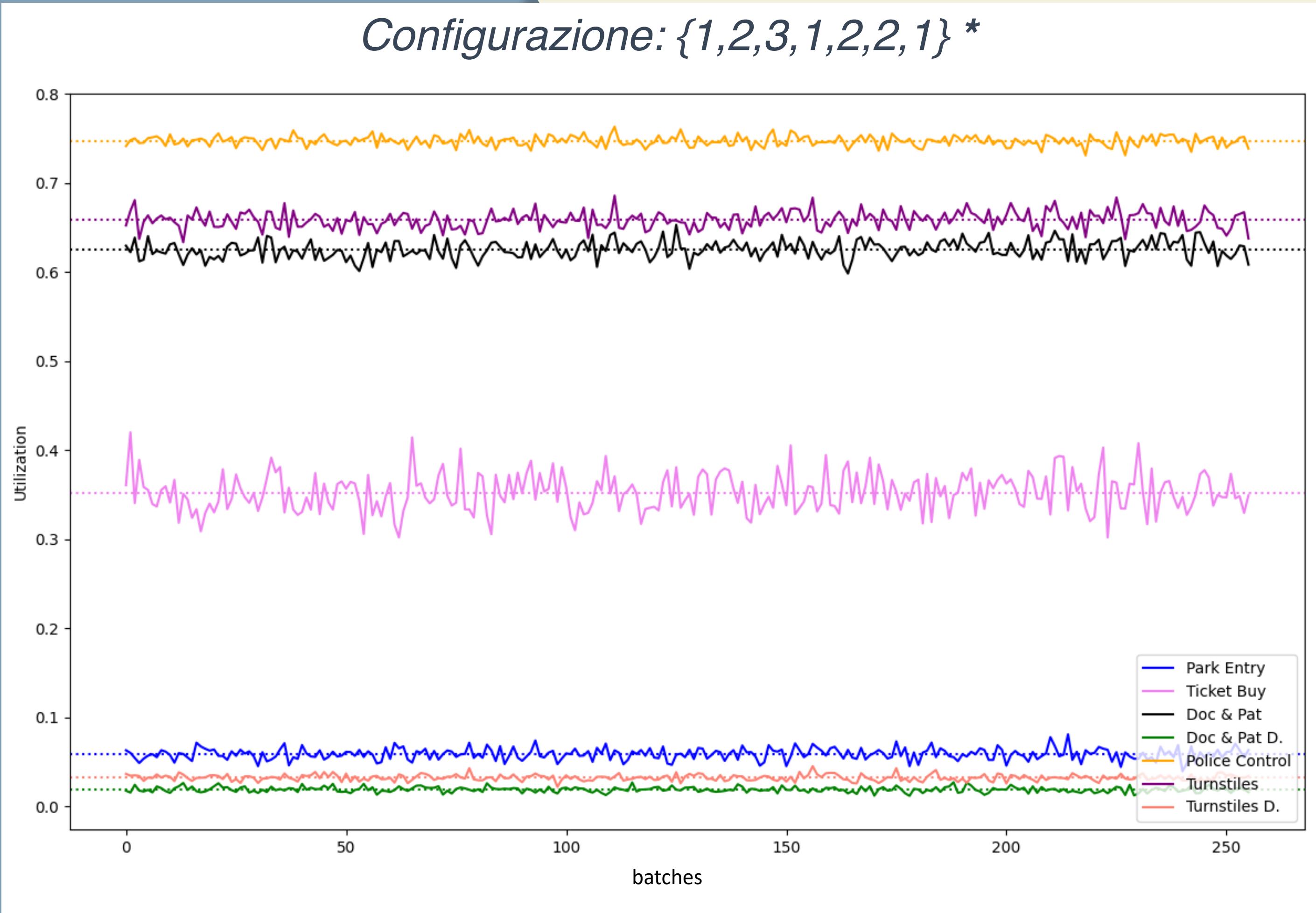
- ▶ Viene rispettato anche il **QoS** che richiede una percentuale minima di controllo del **25%** da parte della polizia
- ▶ La media è circa del **42.5%** ed è quindi il **minimo numero di poliziotti** per rispettare il QoS e non variare il modello

* Ridimensionamento terza fascia:
 $\{1,2,2,1,2,2,1\} \rightarrow \{1,2,3,1,2,2,1\}$

INFINITE HORIZON SIMULATION

3° Fascia 15:00 – 16:00

Configurazione: $\{1,2,3,1,2,2,1\}$ *



Analizzando le utilizzazioni:

- $\{1,2,3,1,2,2,1\}$ è la configurazione minima per la stazionarietà, per non variare il modello (multi-server → single-server) e per rispettare il QoS del tempo di risposta

FINITE HORIZON SIMULATION

La simulazione ad **orizzonte finito** è stata effettuata sulle **3 fasce orarie**.

I risultati sono stati mediati su **256 ripetizioni**:

- ▶ ogni ripetizione fornisce statistiche rappresentanti lo **studio del transiente** del sistema
- ▶ l'insieme delle statistiche rappresenta un punto del campione
- ▶ il sistema è **vuoto** ad inizio e fine simulazione

Il sistema viene assunto **dinamico** al cambio fascia:

- ▶ il **numero di serventi** nei centri viene aumentato e/o diminuito
- ▶ il **flusso di arrivo** varia come indicato nel modello delle specifiche

FINITE HORIZON SIMULATION

Per ogni ripetizione, sono state effettuate le misurazioni del **tempo di risposta del sistema ogni 10 secondi** (come somma dei tempi di risposta dei centri), per un totale di **1080 misurazioni per ripetizione**.

Gli **obiettivi** principali della simulazione ad orizzonte finito sono:

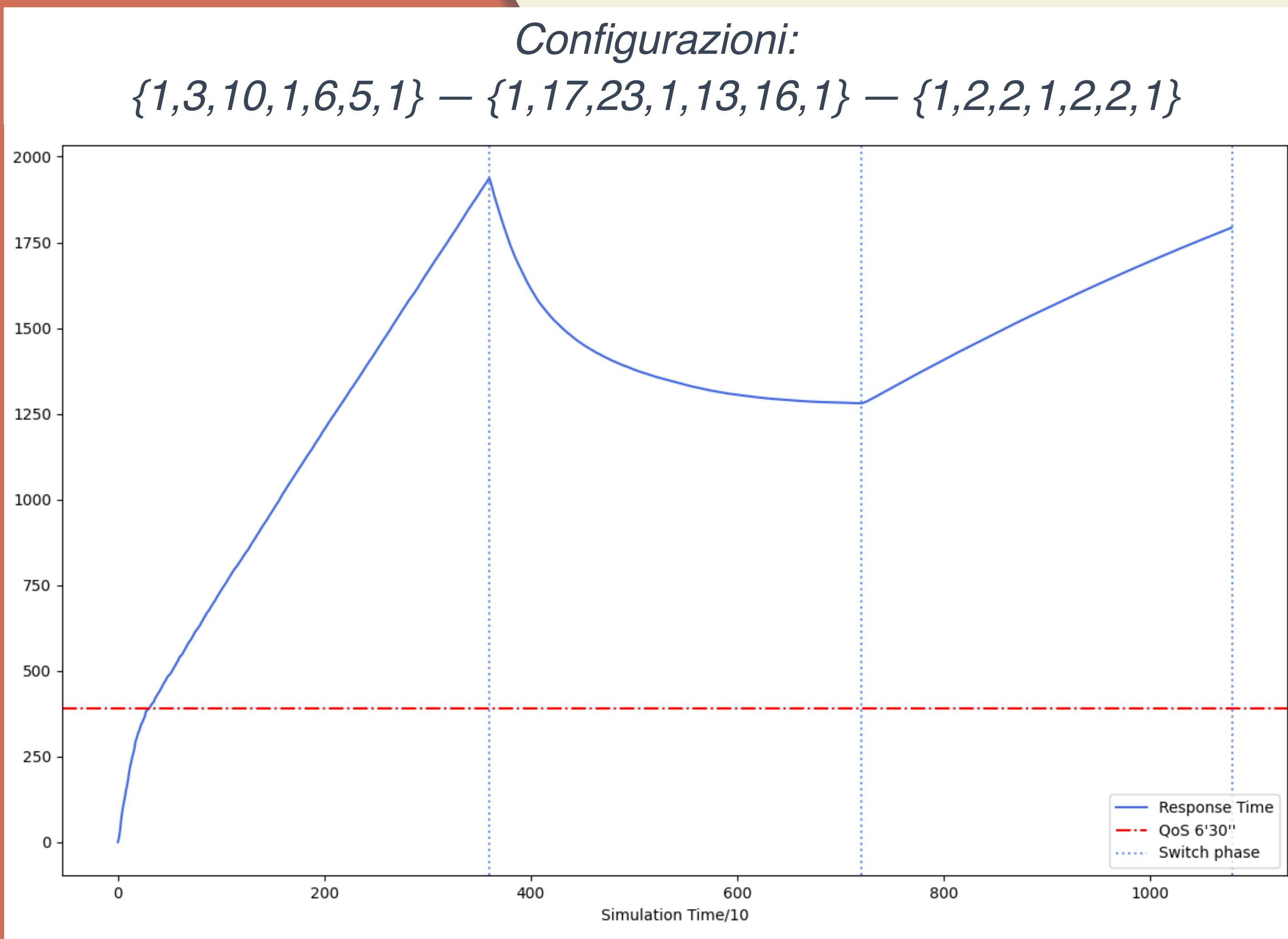
- ▶ analizzare il comportamento del sistema al cambio fascia
- ▶ verificare che le configurazioni individuate siano valide
- ▶ individuare la configurazione migliore per rispettare i QoS e minimizzare i costi
- ▶ analizzare il comportamento del sistema nel transiente

FINITE HORIZON SIMULATION

1

Configurazioni:

$$\{1,3,10,1,6,5,1\} - \{1,17,23,1,13,16,1\} - \{1,2,2,1,2,2,1\}$$



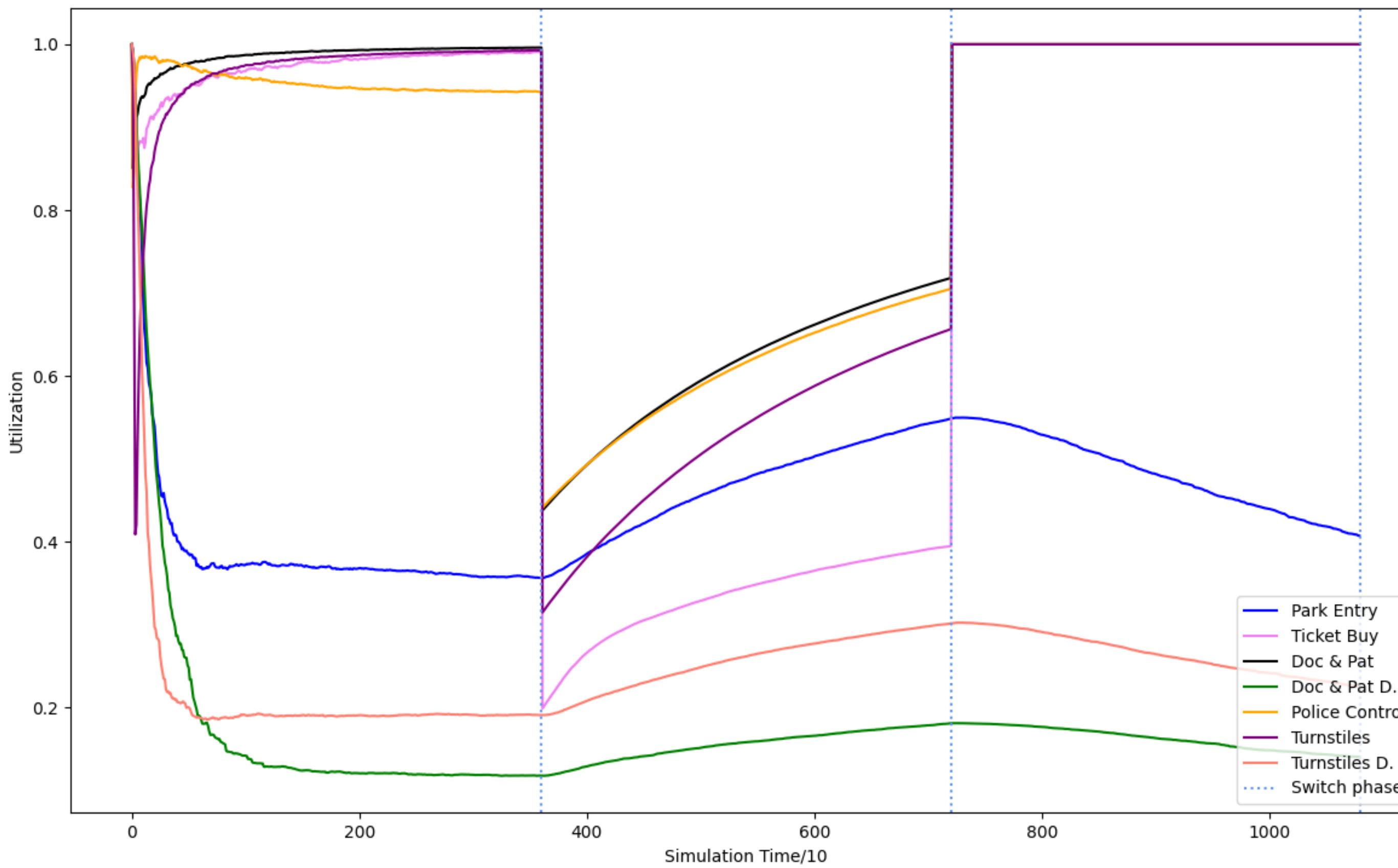
- Il QoS che richiede di avere tempi massimi di risposta del sistema di 6 minuti e mezzo, non viene rispettato
- La prima fascia non riesce a smaltire gli utenti in arrivo
- La seconda fascia inizia in stato di sovraccarico, sembra stabilizzarsi ma sopra l'*upper bound*
- La terza fascia inizia in stato di sovraccarico e non si stabilizza

FINITE HORIZON SIMULATION

1

Configurazioni:

$$\{1,3,10,1,6,5,1\} - \{1,17,23,1,13,16,1\} - \{1,2,2,1,2,2,1\}$$

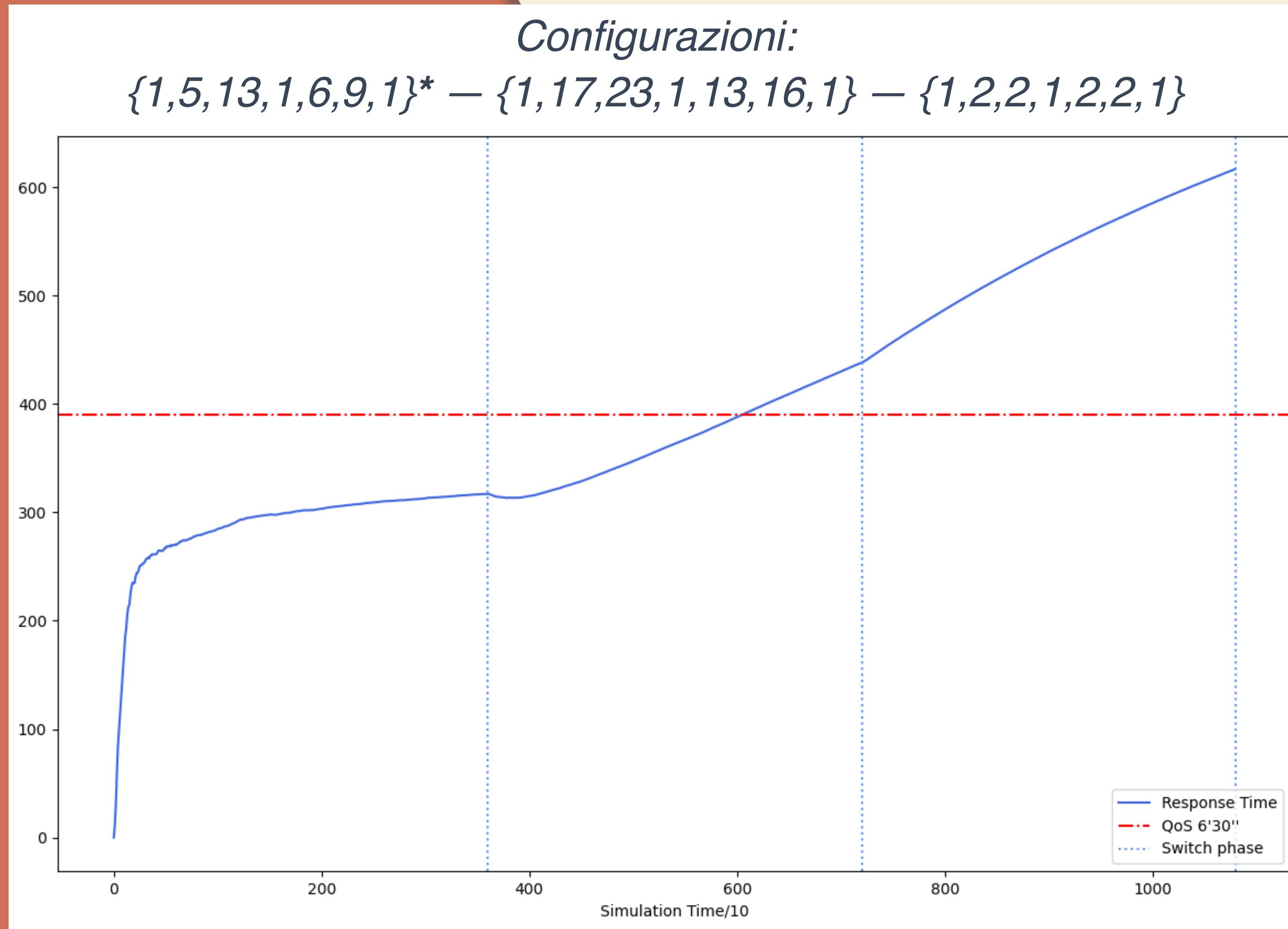


Analizzando le utilizzazioni:

- ▶ nella **prima fascia oraria** c'è *under provisioning* per:
 - Doc & Pat
 - Ticket Buy
 - Turnstiles
- ▶ le successive fasce risultano già sovraccaricate e non valide per l'analisi

FINITE HORIZON SIMULATION

2



* Ridimensionamento prima fascia:
 $\{1,3,10,1,6,5,1\} \rightarrow \{1,5,13,1,6,9,1\}$

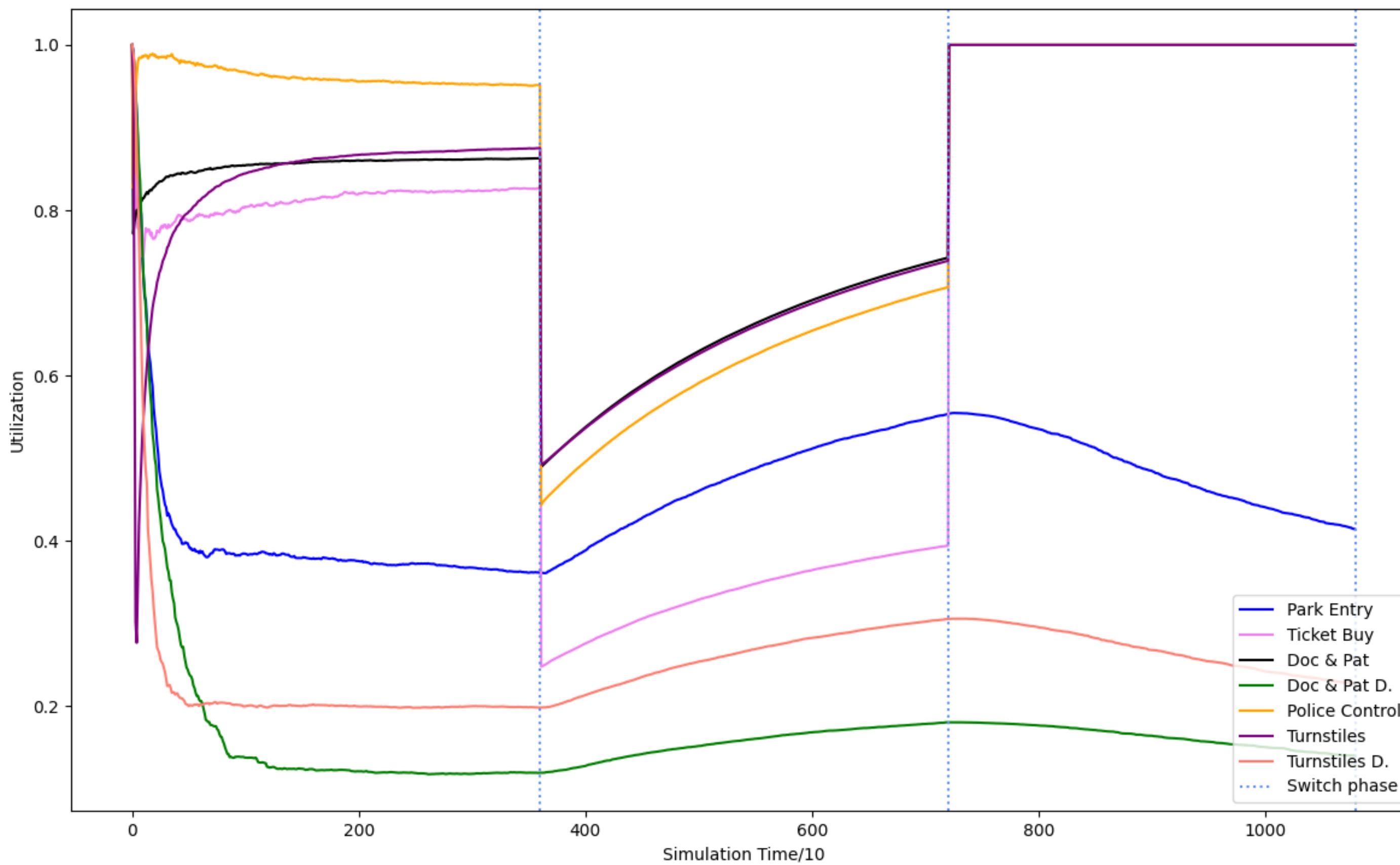
- Il **QoS** che richiede di avere tempi massimi di risposta del sistema di **6 minuti e mezzo**, non viene rispettato
- La **prima fascia** rimane stabile
- La **seconda fascia** non riesce a smaltire gli utenti in arrivo prima dell'*upper bound*
- La **terza fascia** inizia in stato di sovraccarico e non si stabilizza

FINITE HORIZON SIMULATION

2

Configurazioni:

$\{1,5,13,1,6,9,1\}^*$ – $\{1,17,23,1,13,16,1\}$ – $\{1,2,2,1,2,2,1\}$



* Ridimensionamento prima fascia:
 $\{1,3,10,1,6,5,1\} \rightarrow \{1,5,13,1,6,9,1\}$

Analizzando le utilizzazioni:

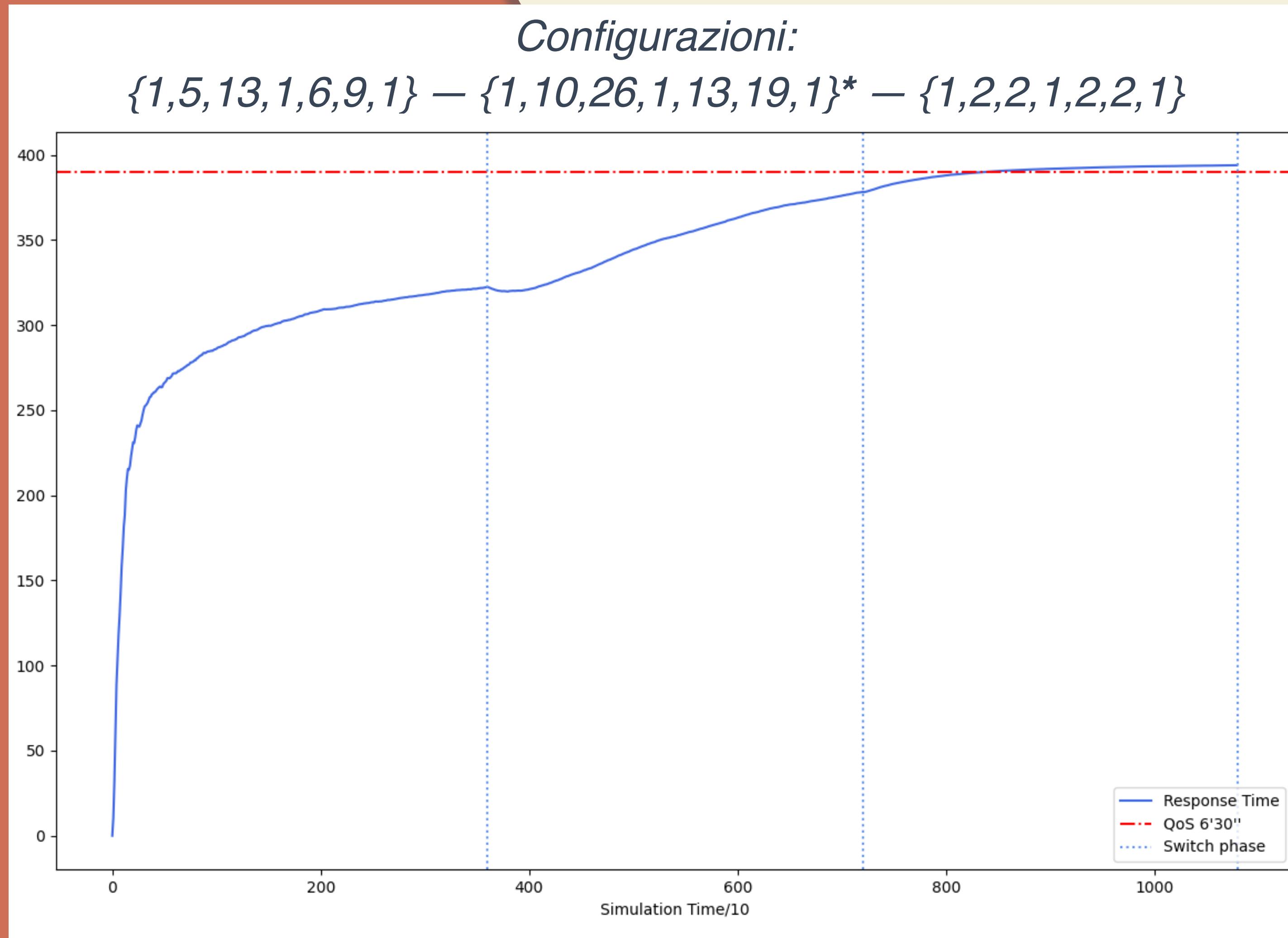
- ▶ la **prima fascia** è stazionaria
- ▶ la **seconda fascia** è stazionaria ma non rispetta il QoS → *ridimensionamento*
- ▶ la **terza fascia** risulta già sovraccarica e non valida per l'analisi

* Ridimensionamento seconda fascia:

$$\{1, 17, 23, 1, 13, 16, 1\} \rightarrow \{1, 10, 26, 1, 13, 19, 1\}$$

FINITE HORIZON SIMULATION

3



- ▶ Il **QoS** che richiede di avere tempi massimi di risposta del sistema di **6 minuti e mezzo**, non viene rispettato
- ▶ La **seconda fascia** si avvicina molto all'*upper bound*
- ▶ La **terza fascia** supera l'*upper bound*
- ▶ Mancano 2 serventi rispetto alla **configurazione ottimale** scaturita dall'analisi ad orizzonte infinito

FINITE HORIZON SIMULATION

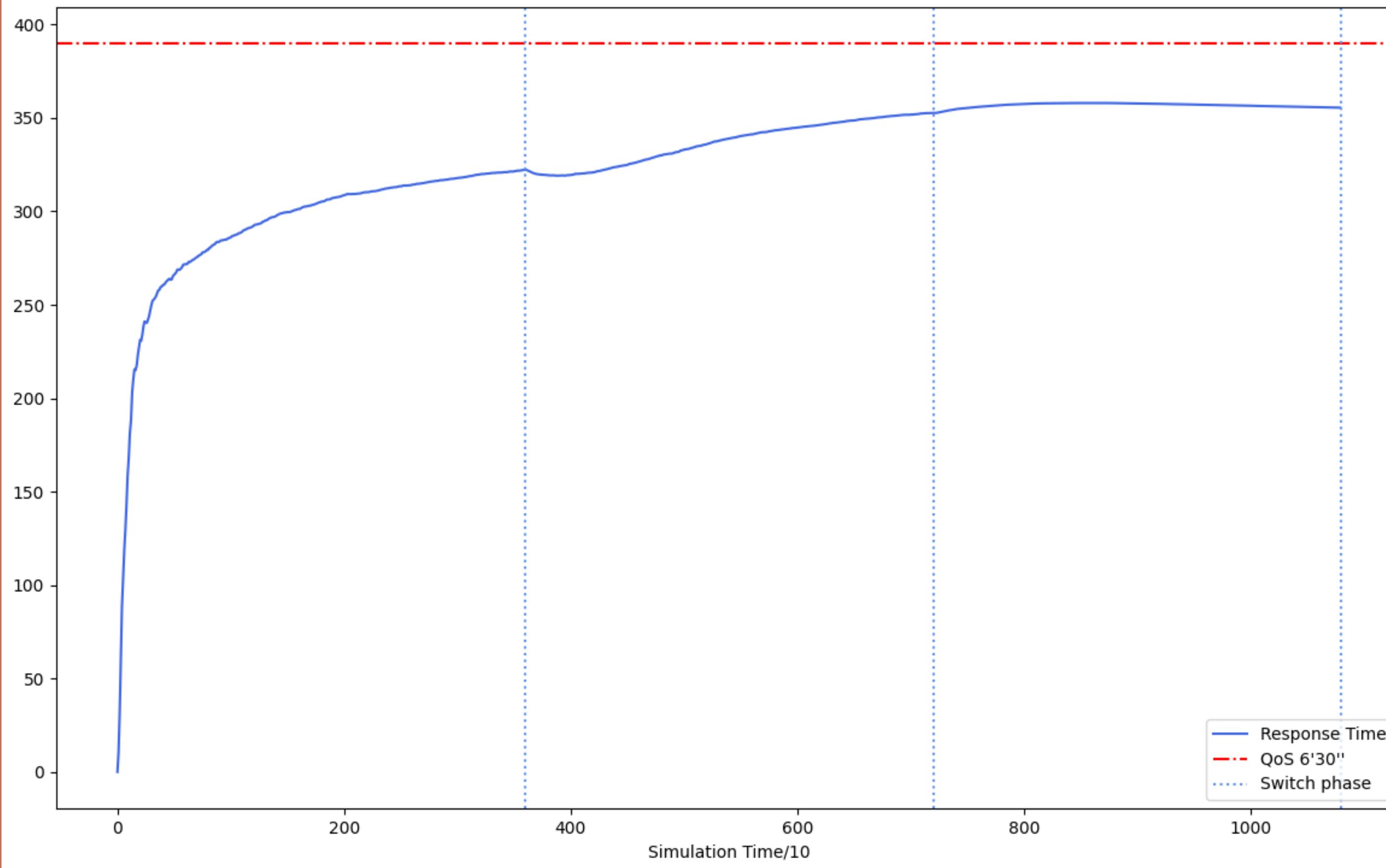
3

* Ridimensionamento seconda fascia:
 $\{1, 10, 26, 1, 13, 19, 1\} \rightarrow \{1, 11, 26, 1, 13, 19, 1\}$

** Ridimensionamento terza fascia:
 $\{1, 2, 2, 1, 2, 2, 1\} \rightarrow \{1, 2, 3, 1, 2, 2, 1\}$

Configurazioni:

$\{1, 5, 13, 1, 6, 9, 1\} - \{1, 11, 26, 1, 13, 19, 1\}^* - \{1, 2, 3, 1, 2, 2, 1\}^{**}$



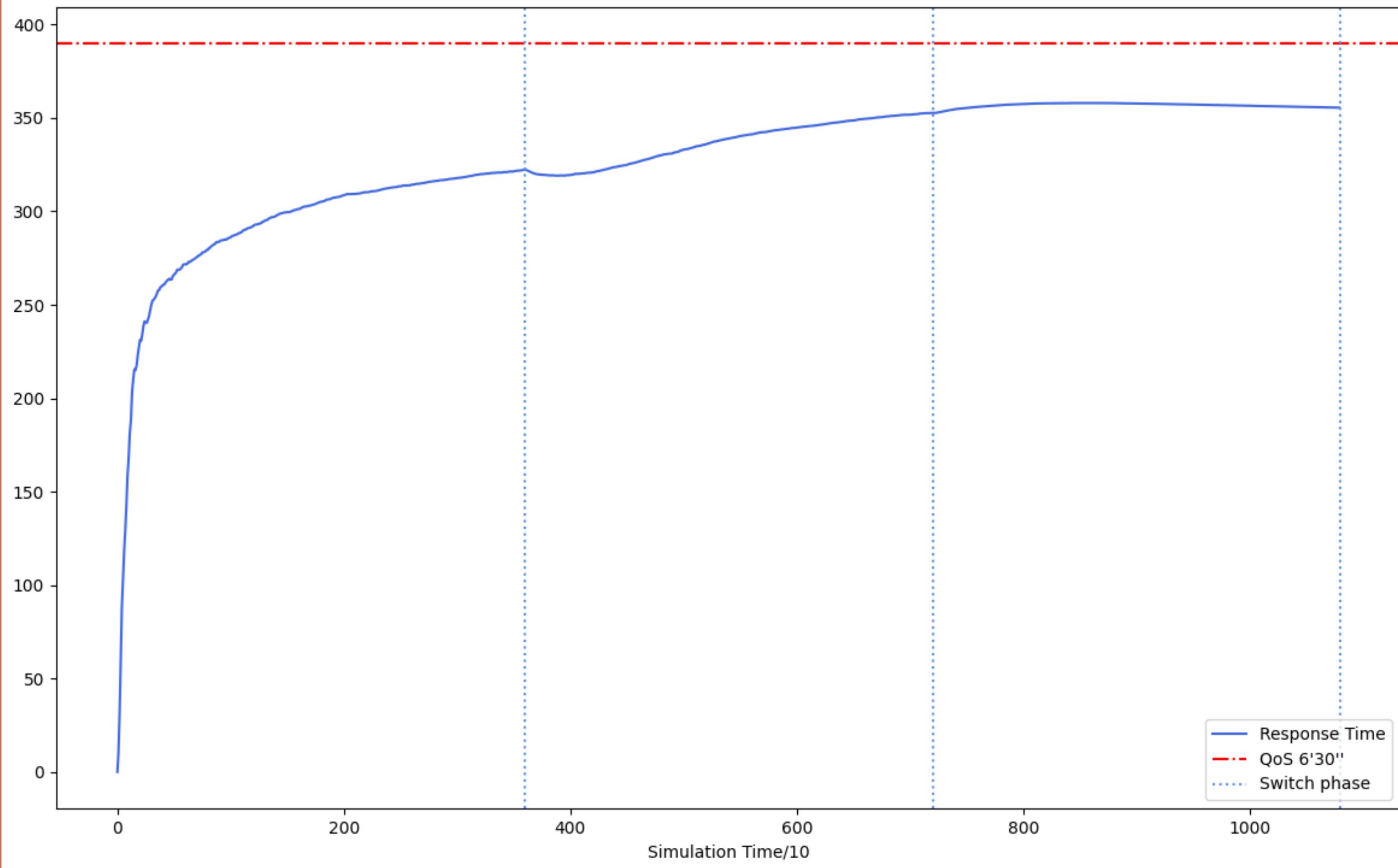
- ▶ Il **QoS** che richiede di avere tempi massimi di risposta del sistema di **6 minuti e mezzo**, viene rispettato
- ▶ Costo totale: **1731€**

FINITE HORIZON SIMULATION

3

Configurazioni:

$\{1,5,13,1,6,9,1\} - \{1,11,26,1,13,19,1\}^* - \{1,2,3,1,2,2,1\}^{**}$



- * Ridimensionamento seconda fascia:
 $\{1,10,26,1,13,19,1\} \rightarrow \{1,11,26,1,13,19,1\}$
- ** Ridimensionamento terza fascia:
 $\{1,2,2,1,2,2,1\} \rightarrow \{1,2,3,1,2,2,1\}$

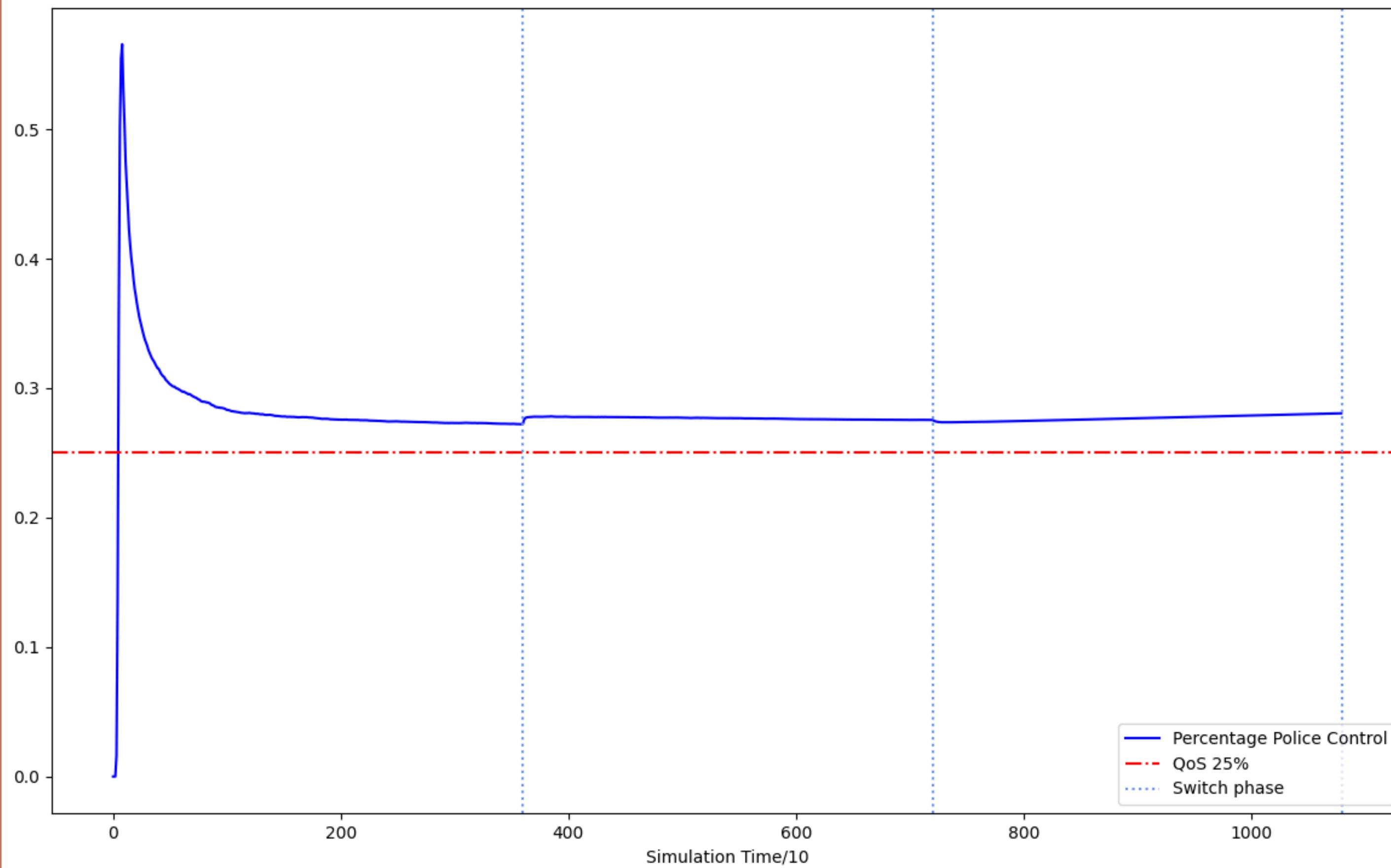
- ▶ Dopo **primo cambio fascia** c'è un decremento del tempo di risposta poiché vengono aggiunti **36 serventi** al sistema
- ▶ Dopo **secondo cambio fascia** c'è un aumento del tempo di risposta poiché vengono rimossi **60 serventi** dal sistema

FINITE HORIZON SIMULATION

3

Configurazioni:

$$\{1,5,13,1,6,9,1\} - \{1,11,26,1,13,19,1\}^* - \{1,2,3,1,2,2,1\}^{**}$$



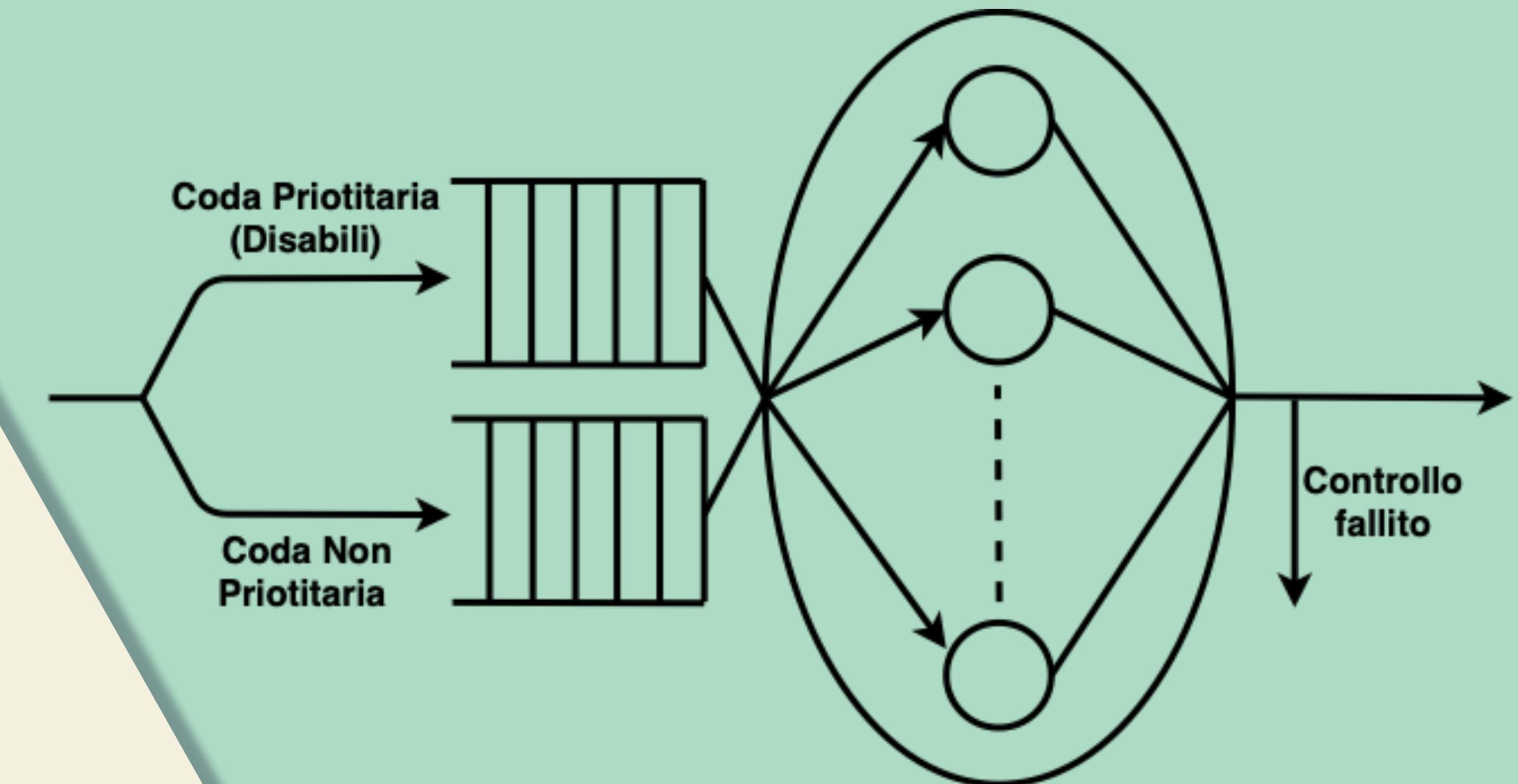
- * Ridimensionamento seconda fascia:
 $\{1,10,26,1,13,19,1\} \rightarrow \{1,11,26,1,13,19,1\}$
- ** Ridimensionamento terza fascia:
 $\{1,2,2,1,2,2,1\} \rightarrow \{1,2,3,1,2,2,1\}$

- ▶ Il **QoS** che richiede una percentuale minima di controllo del **25%** da parte della polizia, viene rispettato
- ▶ La **percentuale di controllo** inizialmente è molto alta poiché i primi utenti verranno controllati tutti essendo il centro ancora vuoto

Il **modello migliorativo** proposto unifica i centri:

- ▶ *Doc & Pat e Doc & Pat D.*
- ▶ *Turnstiles e Turnstiles D.*

e li sostituisce con **2 multiserver** con code a priorità differente senza **preemption ($M/M/m\ NP_priority$)**, pur mantenendo il differente percorso per i disabili.

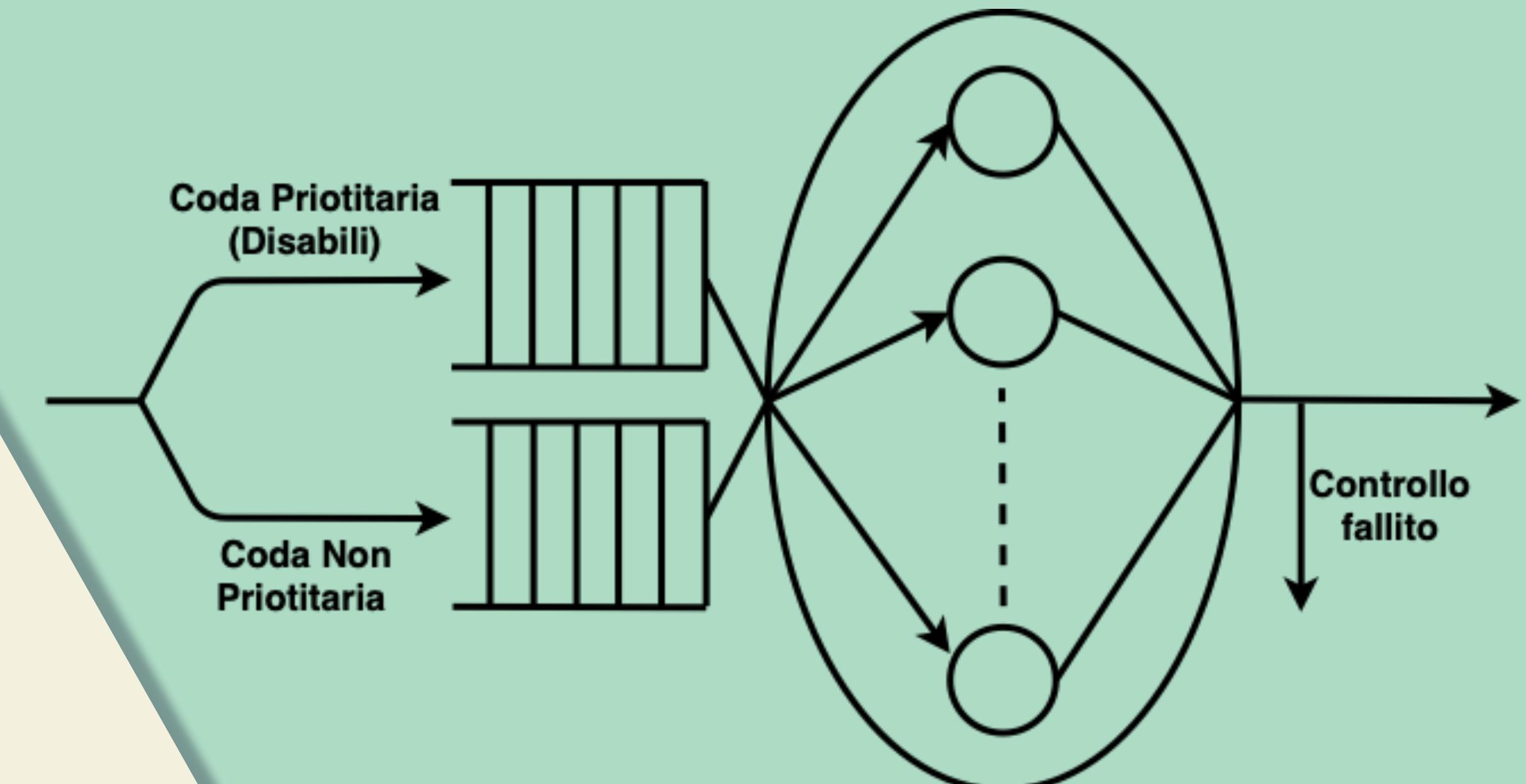


MODELLO MIGLIORATIVO

Obiettivi

Gli obiettivi rimangono:

- ▶ minimizzare i costi
- ▶ rispettare i QoS:
 - avere tempi massimi di risposta del sistema di **6 minuti e mezzo**
 - avere una copertura di controlli della polizia almeno del **25%**

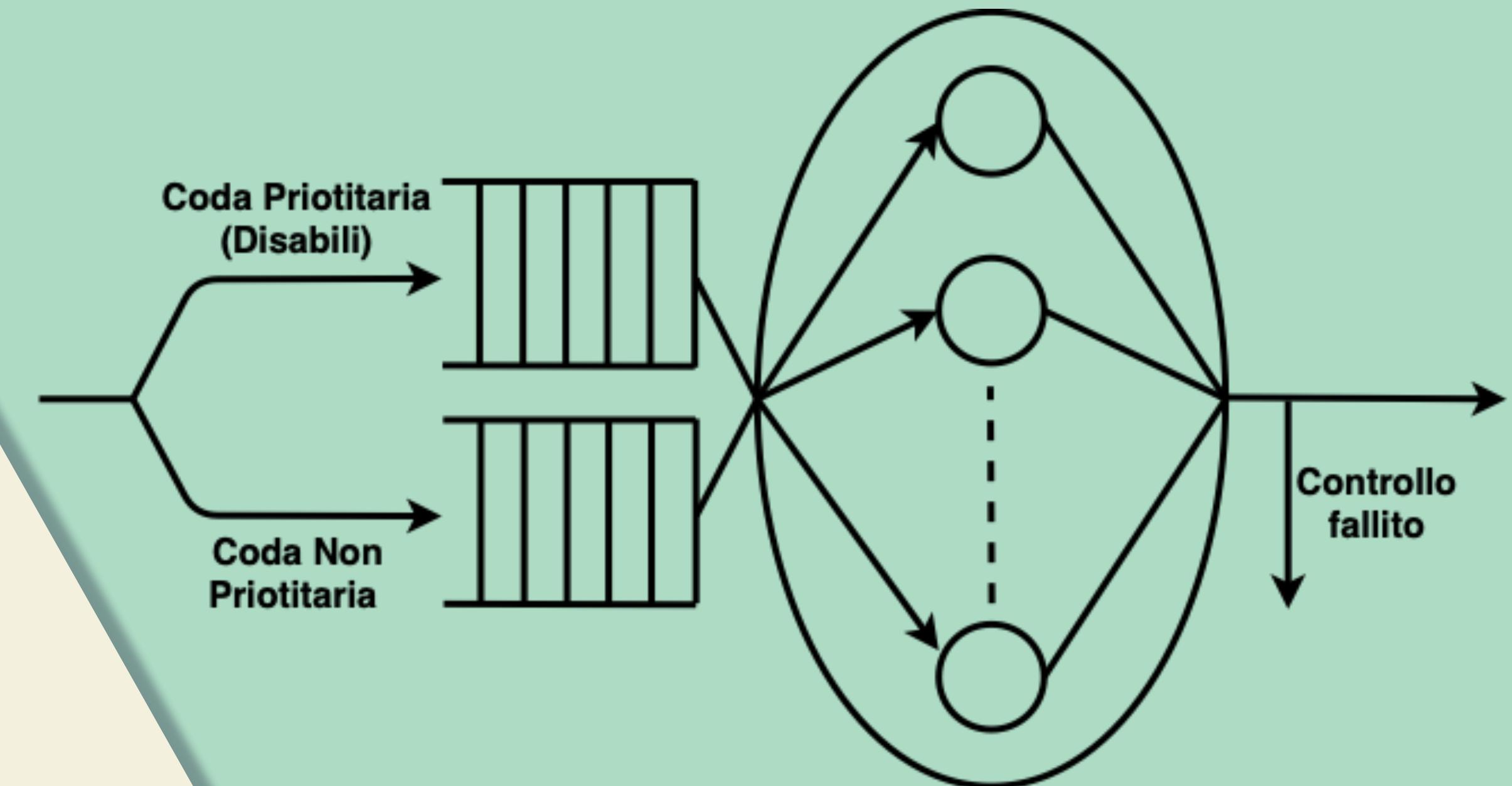


MODELLO MIGLIORATIVO

Obiettivi

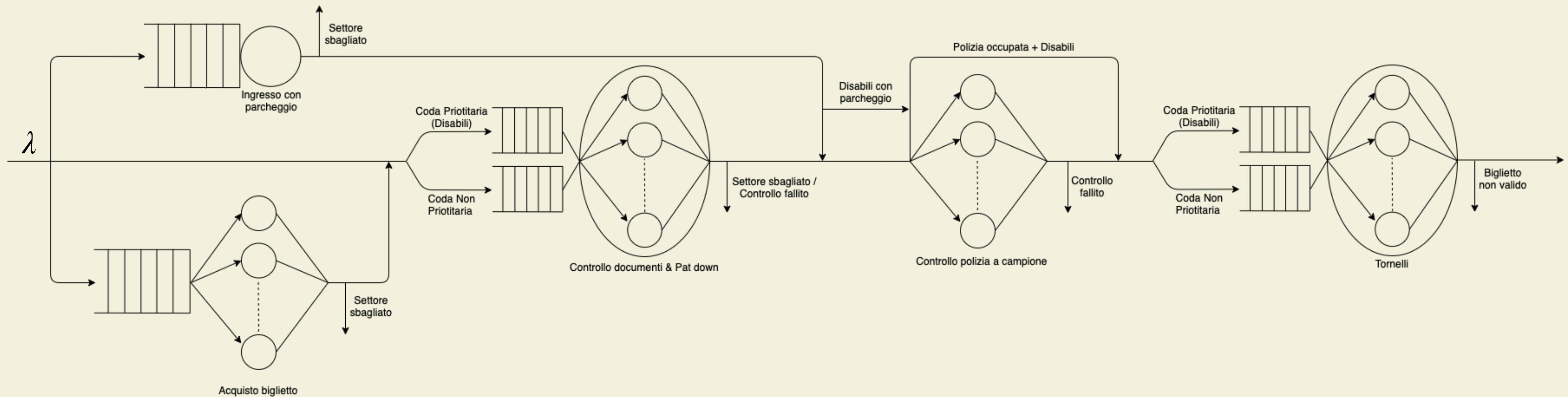
In aggiunta, il nuovo modello permette di soddisfare una delle seguenti caratteristiche:

- ▶ mantenere lo **stesso numero di serventi** totali del modello precedente e **migliorare il tempo di risposta** del sistema
- ▶ **diminuire il numero di serventi** rispetto al modello precedente, mantenere gli **stessi tempi di risposta** ma diminuendo i costi



MODELLO MIGLIORATIVO

Modello Concettuale



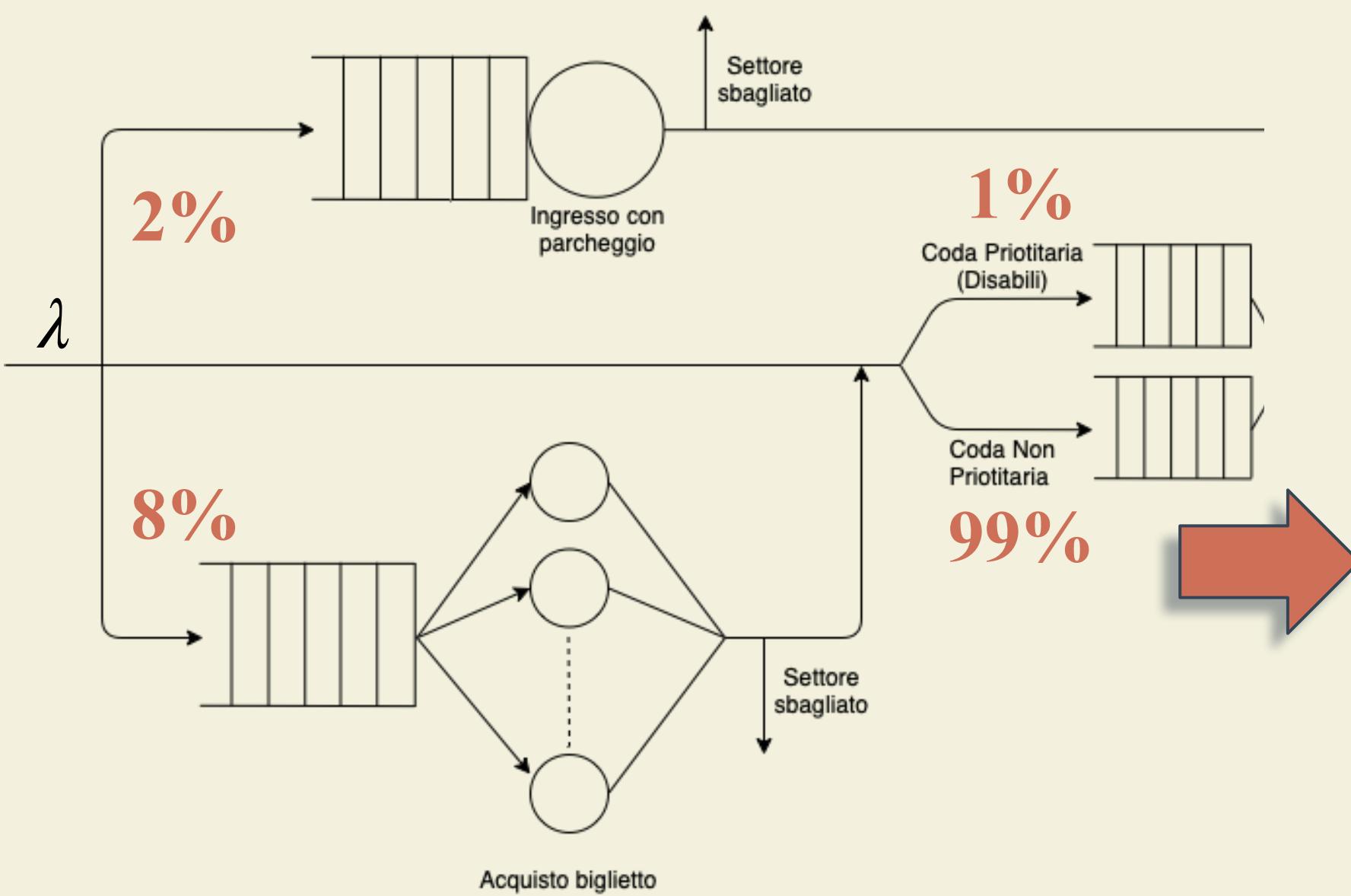
Centri modellati

- ▶ Park Entry ($M/M/1$)
- ▶ Ticket Buy ($M/M/m$)
- ▶ Police Control ($M/M/m/m$)
- ▶ Doc & Pat ($M/M/m\ NP_priority$)
- ▶ Turnstiles ($M/M/m\ NP_priority$)

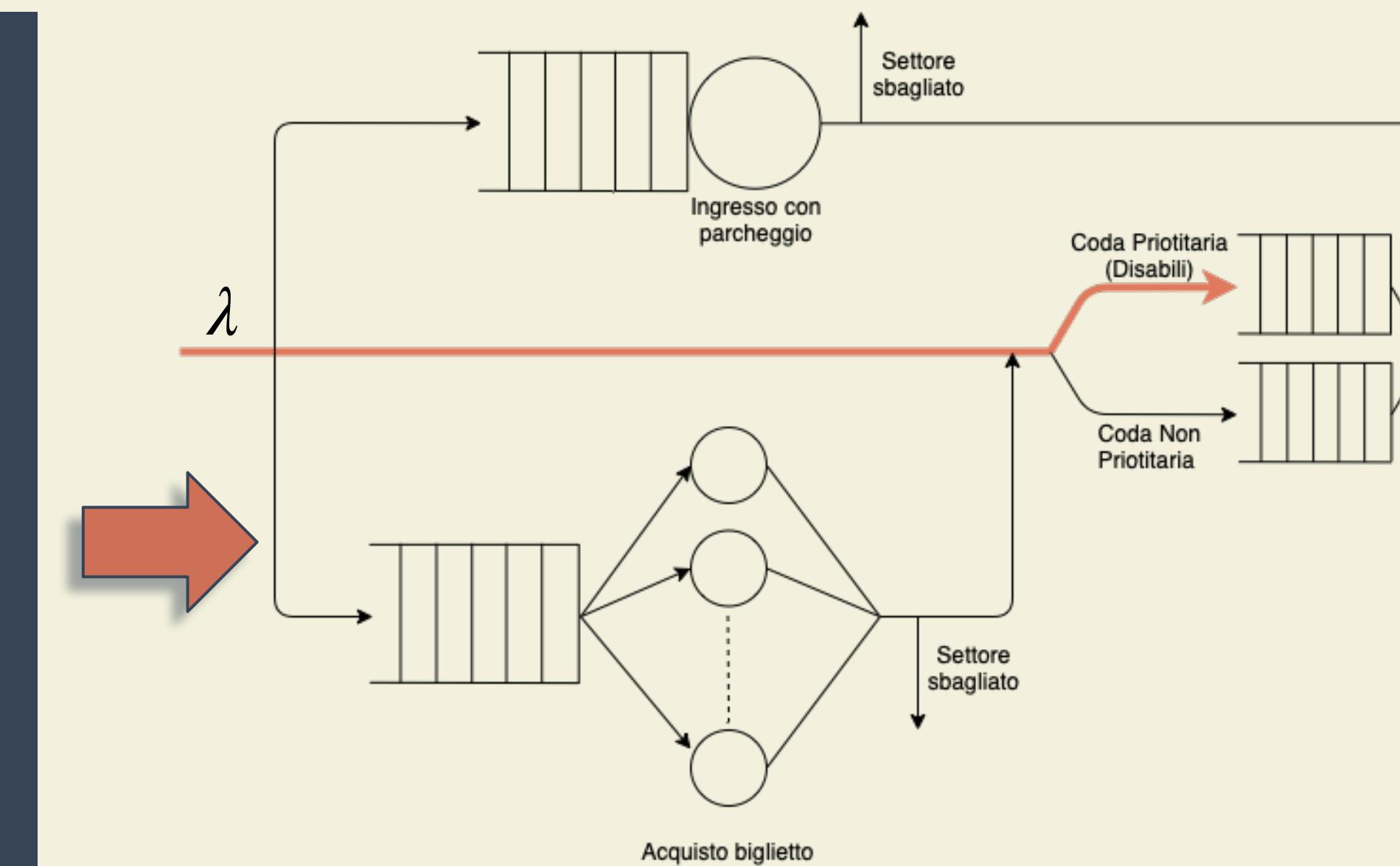
MODELLO MIGLIORATIVO

Modello Computazionale

Nei centri **multi-server con priorità**, per instradare i diversi tipi di utenti sulla giusta coda/percorso, sono state utilizzate le **probabilità di routing** indicate nel modello delle specifiche (rimasto invariato), utilizzando la funzione ***Uniform(0, 1)*** presente nel file *rvgs.c*.



```
x = Uniform(0,1);
if (x <= 0.02) {
    routing PARK_ENTRY;
} else if (x <= 0.10) {
    routing TICKET_BUY;
} else if (x <= 0.11) {
    routing DOC_PAT_prio;
} else {
    routing DOC_PAT_no_prio;
}
```



Per assicurare che i risultati siano conformi alle specifiche, sono stati osservati i seguenti criteri:

- ▶ Il **tempo di risposta** deve essere uguale alla somma del **tempo trascorso in coda** più il **tempo di servizio**
- ▶ Il **numero di arrivi** deve essere uguale al **numero di completamenti** più il **numero di *dropped***
- ▶ Il **numero di utenti in ingresso** in un centro è determinato dalla **probabilità di routing** indicata nel modello delle specifiche

MODELLO MIGLIORATIVO

Verifica - Doc & Pat (MSQ - M/M/27)

- ▶ $E(T_{S_P}) = E(T_{Q_P}) + E(S) =$
 $= 0.54189 + 21.56456 = 22.10646$
- ▶ $E(T_{S_{N_P}}) = E(T_{Q_{N_P}}) + E(S) =$
 $= 4.89679 + 20.10278 = 24.99957$
- ▶ Arrivi = Completamenti + Dropped =
 $= 4199 + 270 = 4469$
- ▶ #expected_arrivals = 4436.344 **

** il calcolo è stato effettuato con lo script 'population.py'

```
for 4199 jobs the service node statistics:  
# arrivals ..... = 4469  
# dropped ..... = 270  
# completions .... = 4199  
  
PRIO NO-PRIOS  
processed ..... = 41 4428  
avg interarrivals .. = 87.80966 0.81305  
avg # in node ..... = 0.24032 29.35080  
avg # in queue .... = 0.00634 5.74908  
avg utilization .... = 0.00934 0.87414  
avg wait ..... = 22.10646 24.99957  
avg delay ..... = 0.54189 4.89679  
avg service ..... = 21.56456 20.10278
```

MODELLO MIGLIORATIVO

Verifica - Turnstiles (MSQ - M/M/20)

- ▶ $E(T_{S_P}) = E(T_{Q_P}) + E(S) =$
 $= 0.38212 + 13.74516 = 14.12728$
- ▶ $E(T_{S_{N_P}}) = E(T_{Q_{N_P}}) + E(S) =$
 $= 4.43330 + 15.00704 = 19.44034$
- ▶ Arrivi = Completamenti + Dropped =
 $= 4229 + 13 = 4242$
- ▶ #expected_arrivals = 4252.71843 **

** il calcolo è stato effettuato con lo script 'population.py'

```
for 4229 jobs the service node statistics:
# arrivals ..... = 4242
# dropped ..... = 13
# completions .... = 4229
                                                               PRIO      NO-PRIOS
processed ..... = 106          4136
avg interarrivals .. = 37.34415   0.95708
avg # in node ..... = 0.37798   20.29476
avg # in queue .... = 0.01071   4.62815
avg utilization .... = 0.01927   0.78333
avg wait ..... = 14.12728   19.44034
avg delay ..... = 0.38212   4.43330
avg service ..... = 13.74516   15.00704
```

MODELLO MIGLIORATIVO

Validazione

Sono stati confrontati i valori dei nuovi centri a code prioritarie con i loro relativi nel modello precedente:

Configurazione: {1, 11, 27, 13, 20}

Doc & Pat

Configurazione: {1, 11, 26, 1, 13, 19, 1}

DOC & PAT (PRIO)

```
# tot. arrivals ..... = 4469
# prio arrivals ..... = 41
# no-prio arrivals .. = 4428
# dropped ..... = 270
# completions ..... = 4199
avg delay prio ..... = 0.41237
avg delay no-prio ... = 4.66989
```

DOC & PAT

```
# arrivals ..... = 4307
# dropped ..... = 213
# completions ..... = 4094
avg delay ..... = 8.11687
```

DOC & PAT D.

```
# arrivals ..... = 29
# dropped ..... = 2
# completions ..... = 27
avg delay ..... = 6.16607
```

MODELLO MIGLIORATIVO

Validazione

Sono stati confrontati i valori dei nuovi centri a code prioritarie con i loro relativi nel modello precedente:

Configurazione: {1, 11, 27, 13, 20}

Turnstiles

Configurazione: {1, 11, 26, 1, 13, 19, 1}

TURNSTILES (PRIO)

```
# tot. arrivals ..... = 4242
# prio arrivals ..... = 106
# no-prio arrivals .. = 4136
# dropped ..... = 13
# completions ..... = 4229
avg delay prio ..... = 0.38723
avg delay no-prio ... = 3.30892
```

TURNSTILES

```
# arrivals ..... = 4078
# dropped ..... = 7
# completions ..... = 4071
avg delay ..... = 5.26622
```

TURNSTILES D.

```
# arrivals ..... = 100
# dropped ..... = 1
# completions ..... = 99
avg delay ..... = 10.59737
```

Sono stati confrontati i risultati delle simulazioni per i **multi-server con priorità**, con i valori teorici scaturiti dalle seguenti formule:

$$p(0) = \left[\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1-\rho)} \right]^{-1} \quad P_Q = \frac{(m\rho)^m}{m!(1-\rho)} p(0)$$

$$E(T_{Q_k}) = \frac{P_Q \cdot E(S)}{(1 - \sum_{i=1}^k \rho_i)(1 - \sum_{i=1}^{k-1} \rho_i)}$$

MODELLO MIGLIORATIVO

Configurazioni

Le diverse **configurazioni di serventi** utilizzate per le **256 simulazioni** della validazione, sono quelle con lo stesso numero di serventi totali del modello precedente:

| | Park Entry | Ticket Buy | Doc & Pat | Police Control | Turnstiles |
|-------|------------|------------|-----------|----------------|------------|
| 13:00 | 1 | 5 | 14 | 6 | 10 |
| 14:00 | 1 | 11 | 27 | 13 | 20 |
| 15:00 | 1 | 2 | 4 | 2 | 3 |
| 16:00 | | | | | |

MODELLO MIGLIORATIVO

Fascia 13:00 – 14:00

* il calcolo è stato effettuato con stesso numero totale di serventi del modello precedente e confrontato con lo script ‘queue_time.py’

| Statistica | Risultato Analitico * | Risultato Sperimentale ($\alpha = 0.05$) |
|------------------------------------|-----------------------|---|
| $E(T_{Q_{PARK_ENTRY}})$ | 16.33205 | 15.54437 ± 0.80040 |
| $E(T_{Q_{TICKET_BUY}})$ | 75.52453 | 71.06907 ± 5.97436 |
| $E(T_{Q_{DOC_PAT}})_{PRIO}$ | 0.52382 | 0.52261 ± 0.02395 |
| $E(T_{Q_{DOC_PAT}})_{NO_PRIO}$ | 2.79264 | 2.64942 ± 0.20092 |
| $E(T_{Q_{POLICE_CONTROL}})$ | 0.00000 | 0.00000 ± 0.00000 |
| $E(T_{Q_{TURNSTILES}})_{PRIO}$ | 0.68044 | 0.69890 ± 0.01927 |
| $E(T_{Q_{TURNSTILES}})_{NO_PRIO}$ | 3.73337 | 3.71195 ± 0.12083 |

MODELLO MIGLIORATIVO

Fascia 14:00 – 15:00

* il calcolo è stato effettuato con stesso numero totale di serventi del modello precedente e confrontato con lo script ‘queue_time.py’

| Statistica | Risultato Analitico * | Risultato Sperimentale ($\alpha = 0.05$) |
|------------------------------------|-----------------------|---|
| $E(T_{Q_{PARK_ENTRY}})$ | 97.02908 | 97.06647 ± 0.96097 |
| $E(T_{Q_{TICKET_BUY}})$ | 77.57197 | 78.53709 ± 1.14106 |
| $E(T_{Q_{DOC_PAT}})_{PRIO}$ | 0.40775 | 0.41379 ± 0.01132 |
| $E(T_{Q_{DOC_PAT}})_{NO_PRIO}$ | 4.67751 | 4.54719 ± 0.13671 |
| $E(T_{Q_{POLICE_CONTROL}})$ | 0.00000 | 0.00000 ± 0.00000 |
| $E(T_{Q_{TURNSTILES}})_{PRIO}$ | 0.3787 | 0.39351 ± 0.02111 |
| $E(T_{Q_{TURNSTILES}})_{NO_PRIO}$ | 3.32145 | 3.35981 ± 0.03857 |

MODELLO MIGLIORATIVO

Fascia 15:00 – 16:00

* il calcolo è stato effettuato con stesso numero totale di serventi del modello precedente e confrontato con lo script ‘queue_time.py’

| Statistica | Risultato Analitico * | Risultato Sperimentale ($\alpha = 0.05$) |
|------------------------------------|-----------------------|---|
| $E(T_{Q_{PARK_ENTRY}})$ | 1.86973 | 1.85882 ± 0.04519 |
| $E(T_{Q_{TICKET_BUY}})$ | 12.7291 | 12.74679 ± 0.17830 |
| $E(T_{Q_{DOC_PAT}})_{PRIO}$ | 0.74704 | 0.82044 ± 0.07356 |
| $E(T_{Q_{DOC_PAT}})_{NO_PRIO}$ | 1.41822 | 1.32311 ± 0.09802 |
| $E(T_{Q_{POLICE_CONTROL}})$ | 0.00000 | 0.00000 ± 0.00000 |
| $E(T_{Q_{TURNSTILES}})_{PRIO}$ | 0.95233 | 0.95908 ± 0.22157 |
| $E(T_{Q_{TURNSTILES}})_{NO_PRIO}$ | 1.74313 | 1.68677 ± 0.07972 |

MODELLO MIGLIORATIVO

Confronto Configurazioni

Stesso numero di serventi totali del modello precedente:

| | PE | TB | DP | PC | T |
|-----------|----|----|----|----|----|
| 1° Fascia | 1 | 5 | 14 | 6 | 10 |
| 2° Fascia | 1 | 11 | 27 | 13 | 20 |
| 3° Fascia | 1 | 2 | 4 | 2 | 3 |

Costo totale: 1731€

Minor numero di serventi totali rispetto al modello precedente:

| | PE | TB | DP | PC | T | |
|-----------|----|----|----|----|----|-----------|
| 1° Fascia | 1 | 5 | 12 | 6 | 9 | 1° Fascia |
| 2° Fascia | 1 | 11 | 26 | 13 | 18 | 2° Fascia |
| 3° Fascia | 1 | 2 | 3 | 2 | 2 | 3° Fascia |

Costo totale: 1631€

Come nel modello precedente, la simulazione ad **orizzonte infinito** è stata effettuata su un lasso di tempo molto più lungo rispetto al tempo reale.

Si considera una singola fascia oraria per volta.

Il sistema viene assunto **statico**:

- ▶ il **numero di serventi** rimane invariato
- ▶ il **flusso di arrivo** rimane invariato

Gli **obiettivi** principali sono:

- ▶ analizzare il comportamento del sistema allo stato stazionario
- ▶ individuare la configurazione migliore per rispettare i **QoS** e **minimizzare i costi**

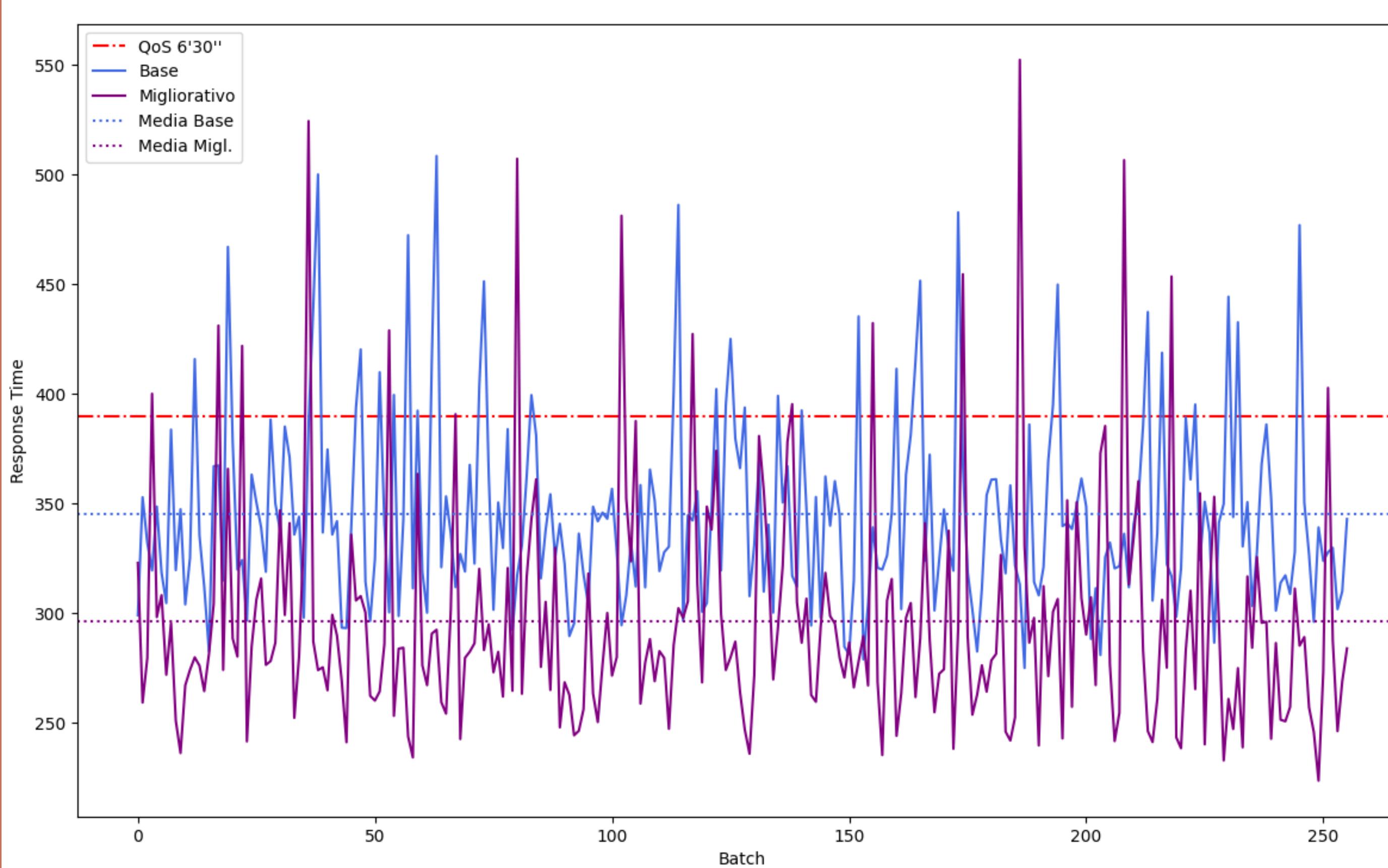
È stato utilizzato il metodo delle *Batch Means*:

- ▶ è stata suddivisa l'esecuzione in $k = 256$ batch di $b = 8192$ jobs
 - k e b scelti usando il metodo di Banks, Carson, Nelson, e Nicol
 - fissato k , si è cercato il valore di b per cui l'*autocorrelazione* del campione sia minore di 0.2 per lag $j=1$
- ▶ sono state calcolate le statistiche per ogni batch
- ▶ è stato generato un campione di k batches indipendenti su cui studiare la media campionaria

MODELLO MIGLIORATIVO

Infinite Horizon Simulation – 1° Fascia

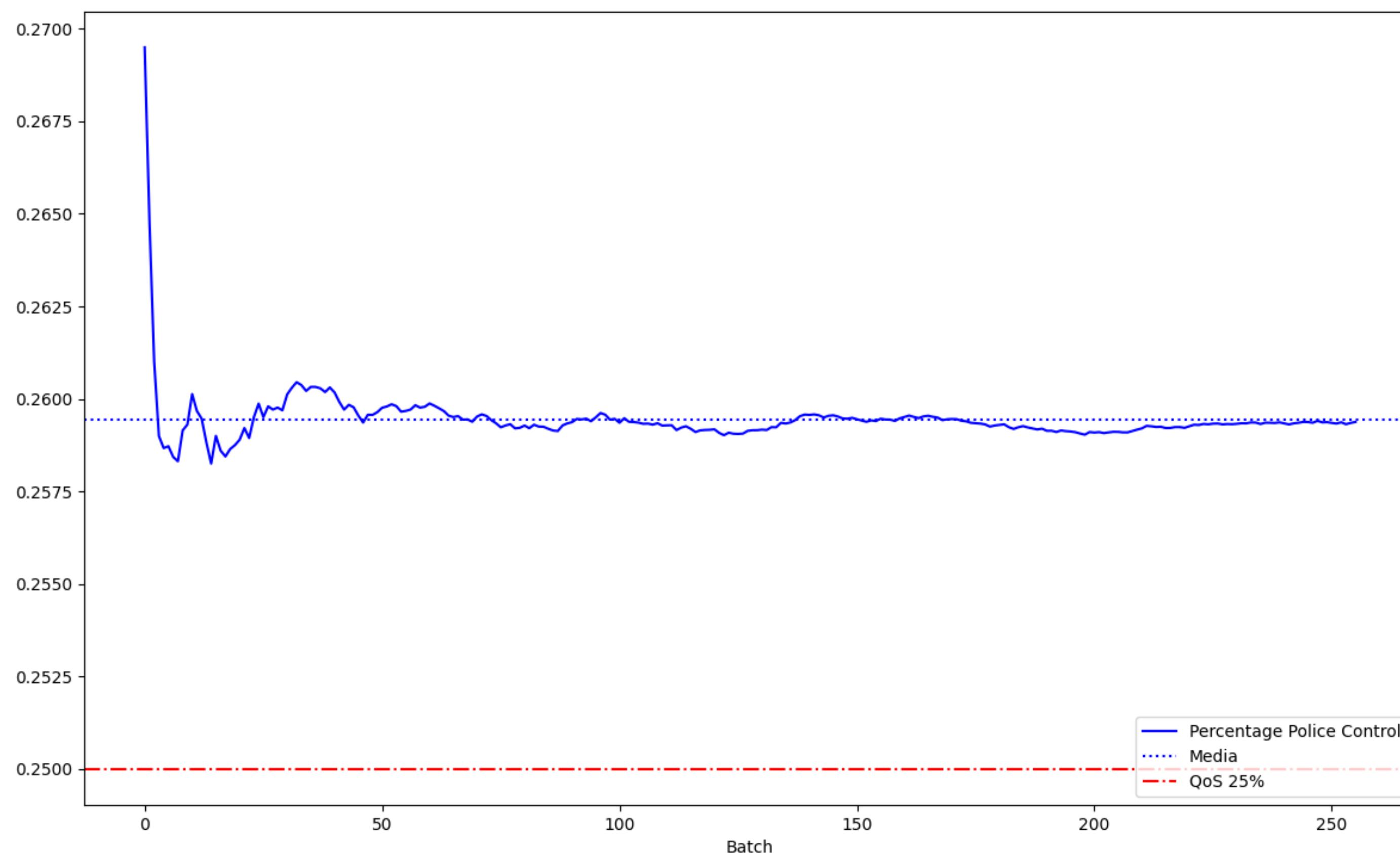
Configurazione: {1,5,14,6,10}



Stesso numero di serventi
totali del modello precedente

- ▶ Il sistema è stazionario
- ▶ Risultato Analitico (da 'queue_time.py'): 293.2873
- ▶ Risultato Sperimentale*: 288.33730 ± 5.72353
- ▶ Rispetta il QoS ed è migliorato rispetto al modello base mediamente di circa 50 secondi

Configurazione: {1,5,14,6,10}



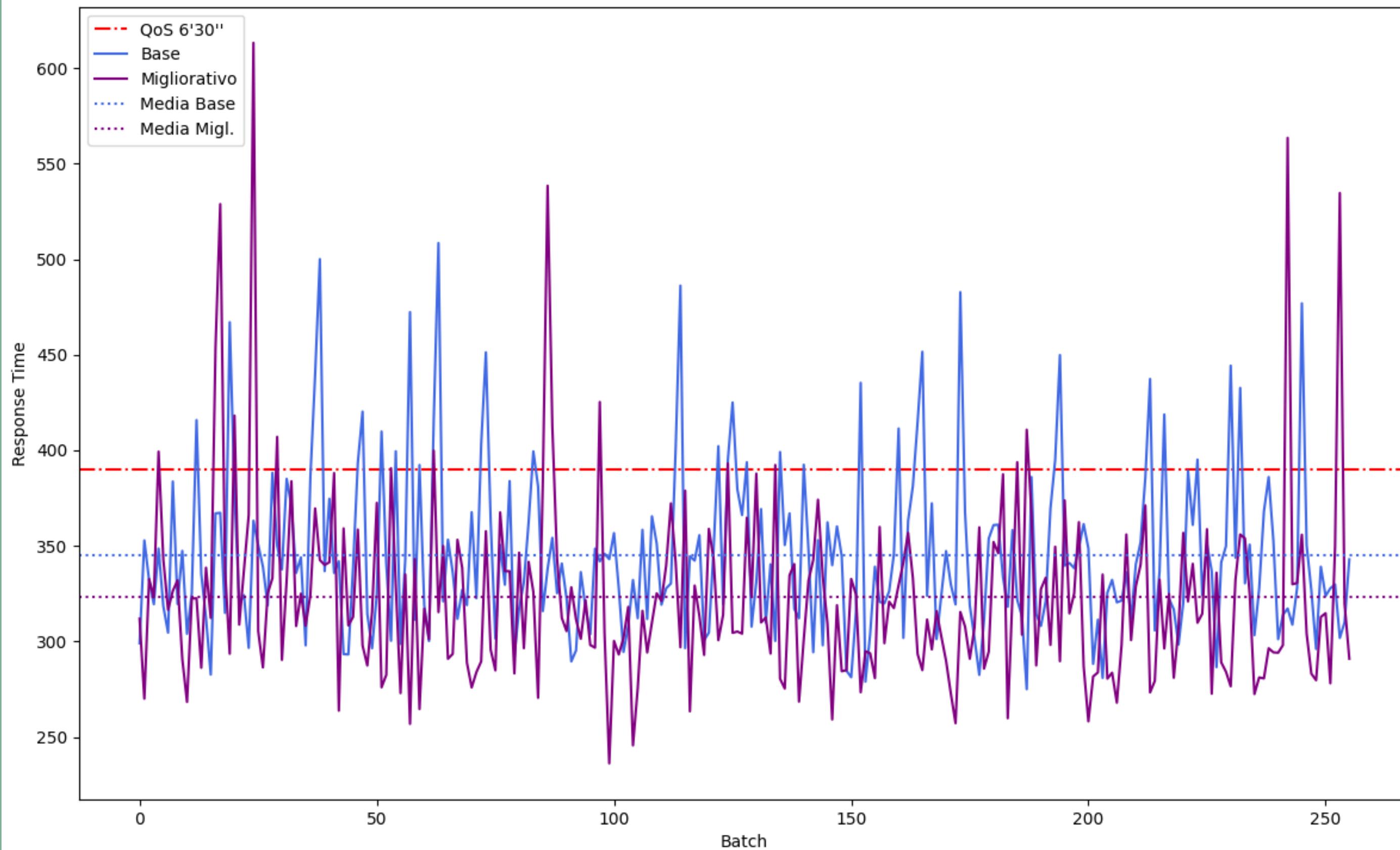
Stesso numero di serventi totali del modello precedente

- ▶ Viene rispettato anche il **QoS** che richiede una percentuale minima di controllo del **25%** da parte della polizia
- ▶ La media è circa del **26%** ed è quindi il **minimo numero di poliziotti** per rispettare il QoS

MODELLO MIGLIORATIVO

Infinite Horizon Simulation – 1° Fascia

Configurazione: {1,5,12,6,9}



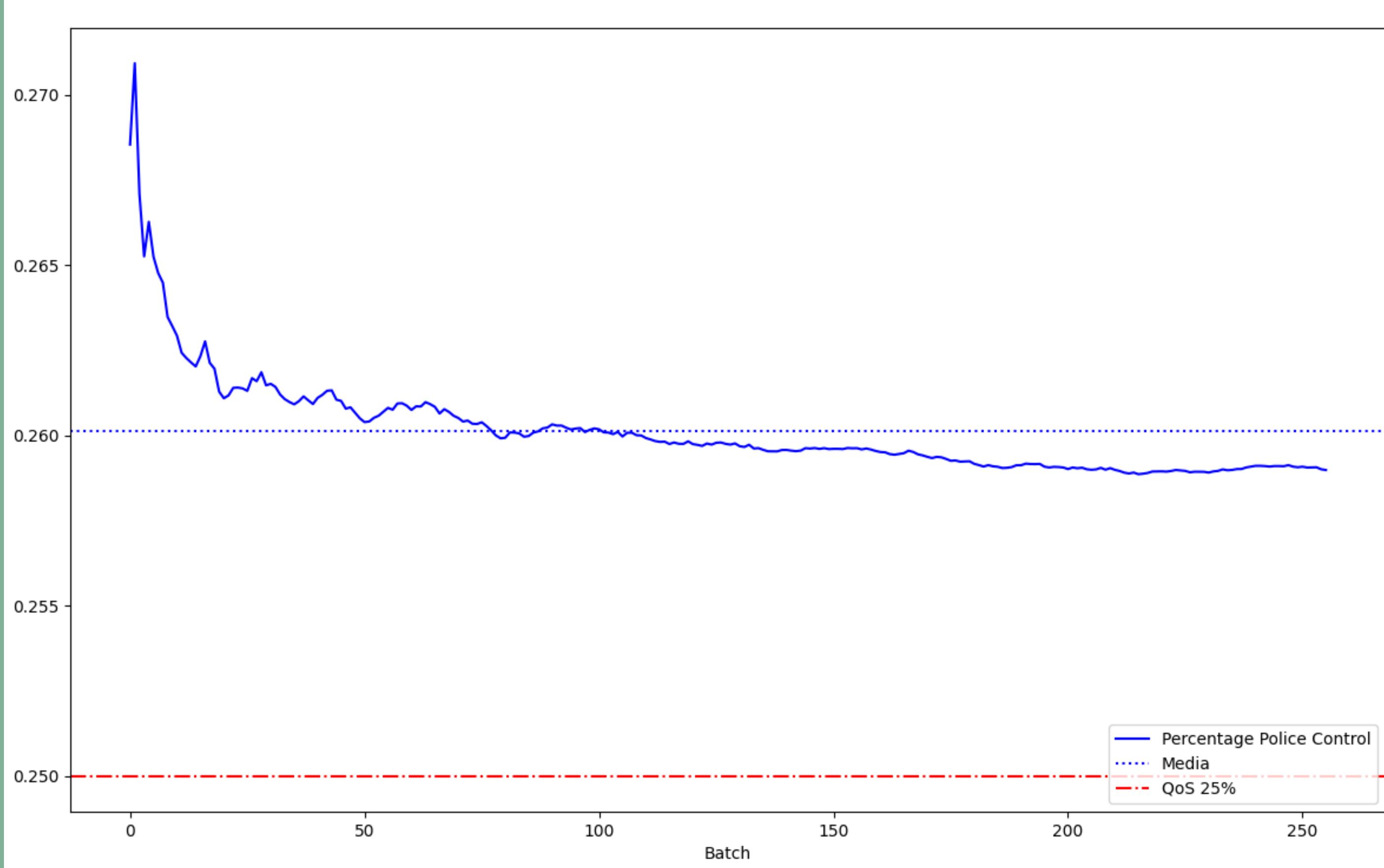
Minor numero di serventi totali del modello precedente

- ▶ Il sistema è **stazionario**
- ▶ Risultato Analitico (da ‘queue_time.py’): **343.61223**
- ▶ Risultato Sperimentale*: **325.51322 ± 7.64107**
- ▶ Rispetta il **QoS** ed è **migliorato** rispetto al modello base mediamente **di circa 18 secondi**

MODELLO MIGLIORATIVO

Infinite Horizon Simulation – 1° Fascia

Configurazione: {1,5,12,6,9}



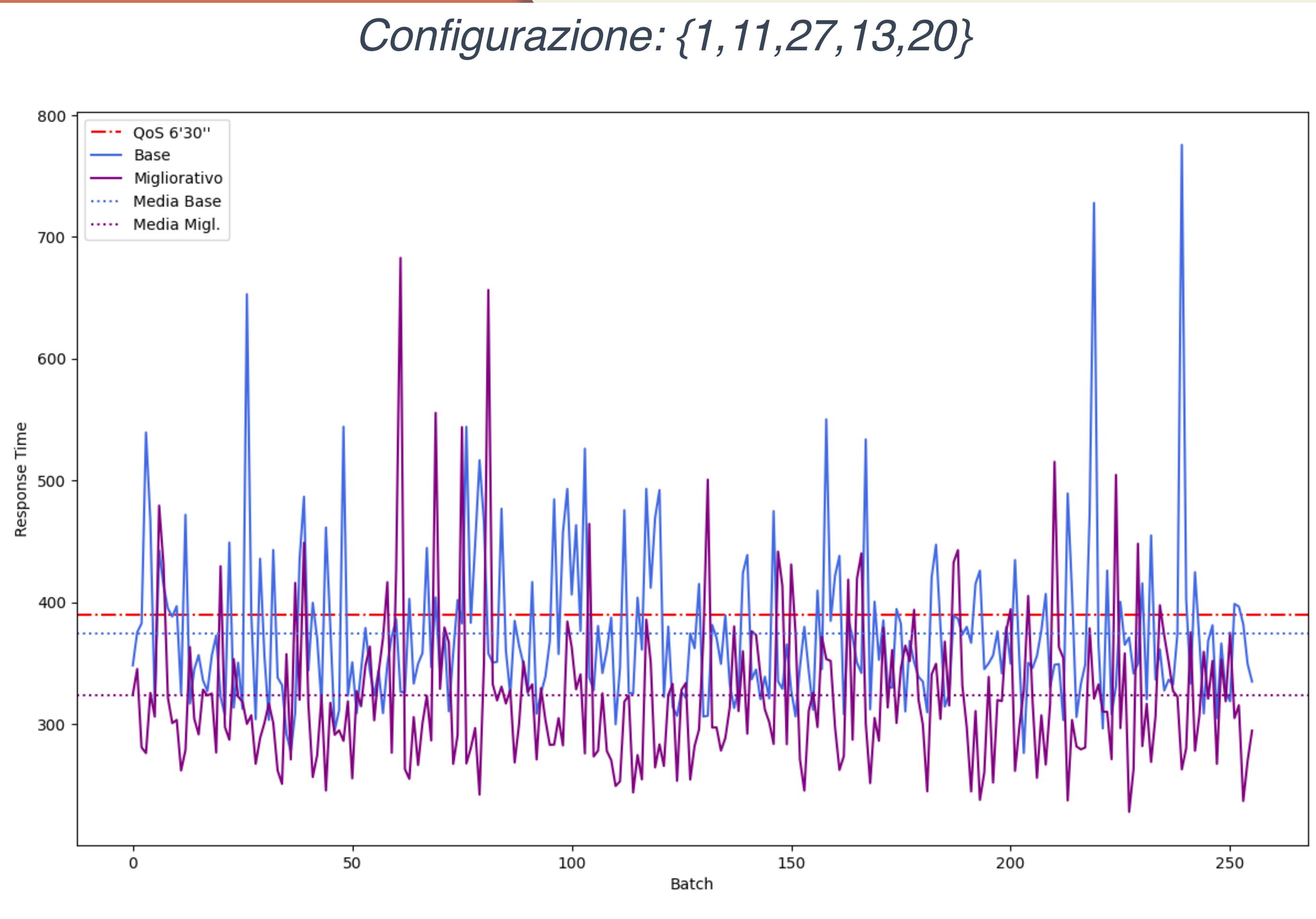
Minor numero di serventi totali del modello precedente

- ▶ Viene rispettato anche il **QoS** che richiede una percentuale minima di controllo del **25%** da parte della polizia
- ▶ La media è circa del **26%** ed è quindi il **minimo numero di poliziotti** per rispettare il QoS

MODELLO MIGLIORATIVO

Infinite Horizon Simulation – 2° Fascia

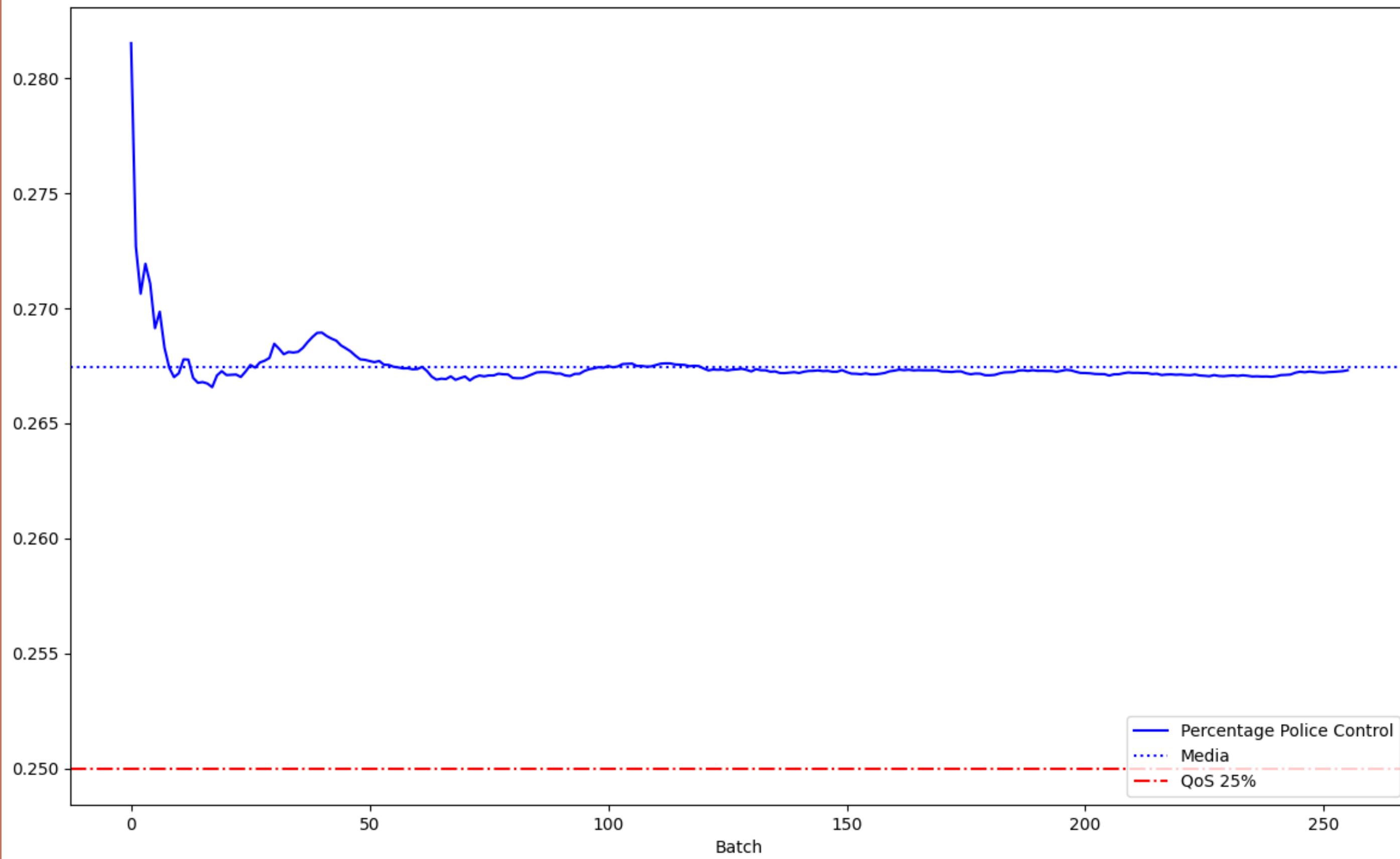
Configurazione: {1, 11, 27, 13, 20}



Stesso numero di serventi
totali del modello precedente

- ▶ Il sistema è stazionario
- ▶ Risultato Analitico (da 'queue_time.py'): 322.8628
- ▶ Risultato Sperimentale*: 325.52347 ± 9.46222
- ▶ Rispetta il QoS ed è migliorato rispetto al modello base mediamente di circa 58 secondi

Configurazione: {1,11,27,13,20}



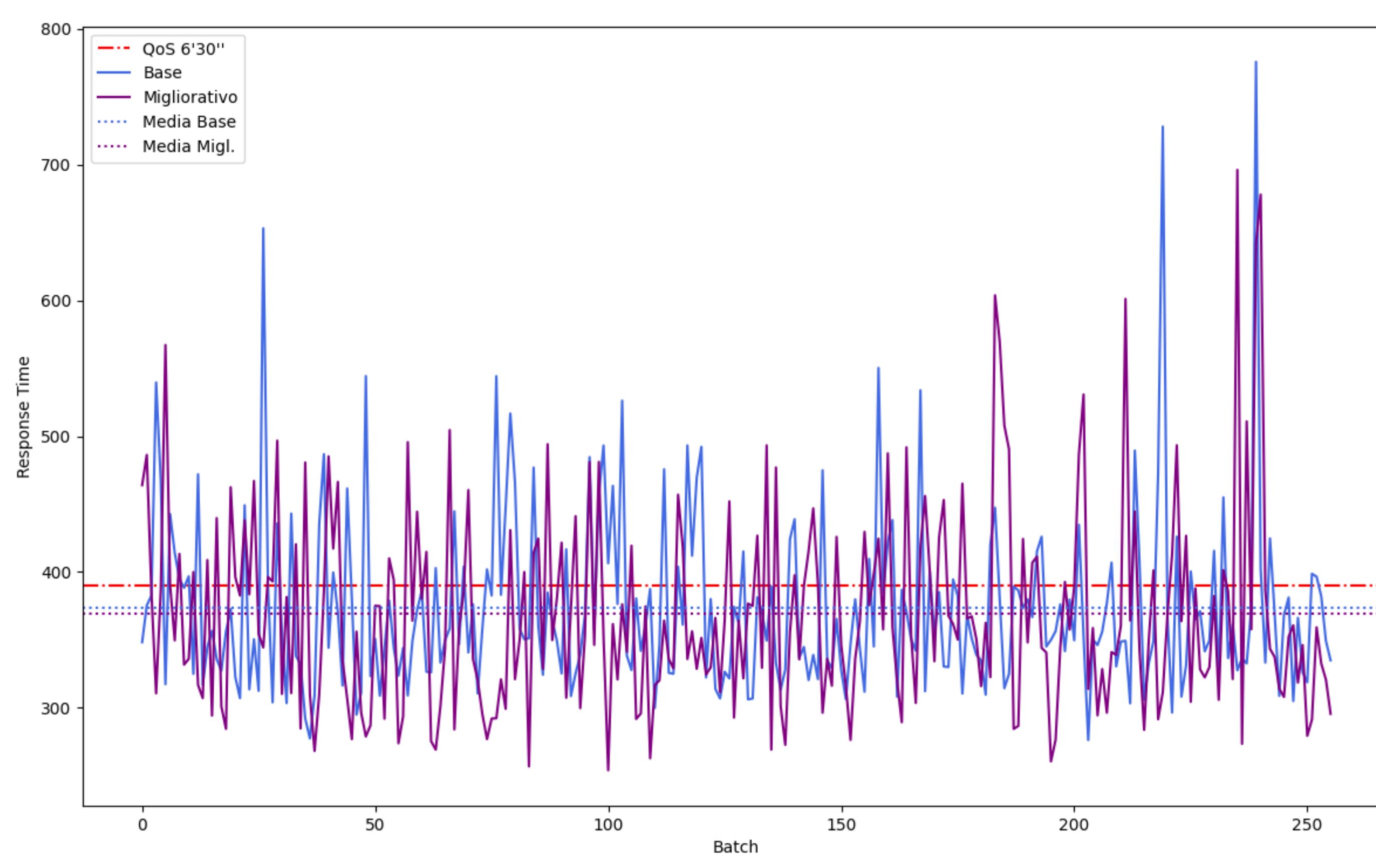
Stesso numero di serventi totali del modello precedente

- ▶ Viene rispettato anche il QoS che richiede una percentuale minima di controllo del 25% da parte della polizia
- ▶ La media è compresa tra il 26.5% e il 27% ed è quindi il minimo numero di poliziotti per rispettare il QoS

MODELLO MIGLIORATIVO

Infinite Horizon Simulation – 2° Fascia

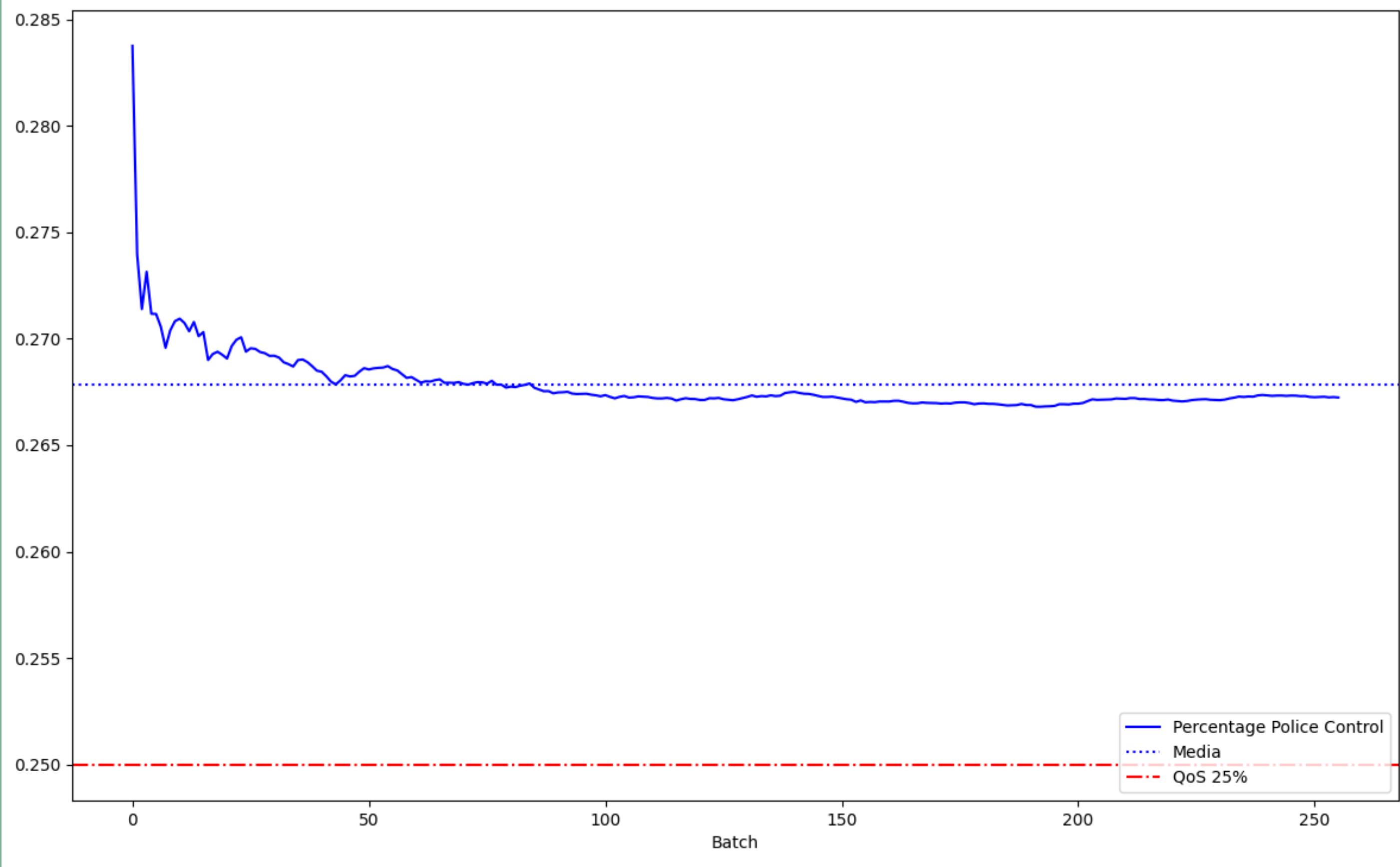
Configurazione: {1, 11, 26, 13, 18}



Minor numero di serventi totali del modello precedente

- ▶ Il sistema è **stazionario**
- ▶ Risultato Analitico (da ‘*queue_time.py*’): **374.262679**
- ▶ Risultato Sperimentale*: **367.91802 ± 9.29505**
- ▶ Rispetta il **QoS** ed è **migliorato** rispetto al modello base mediamente **di circa 6 secondi**

Configurazione: {1, 11, 26, 13, 18}



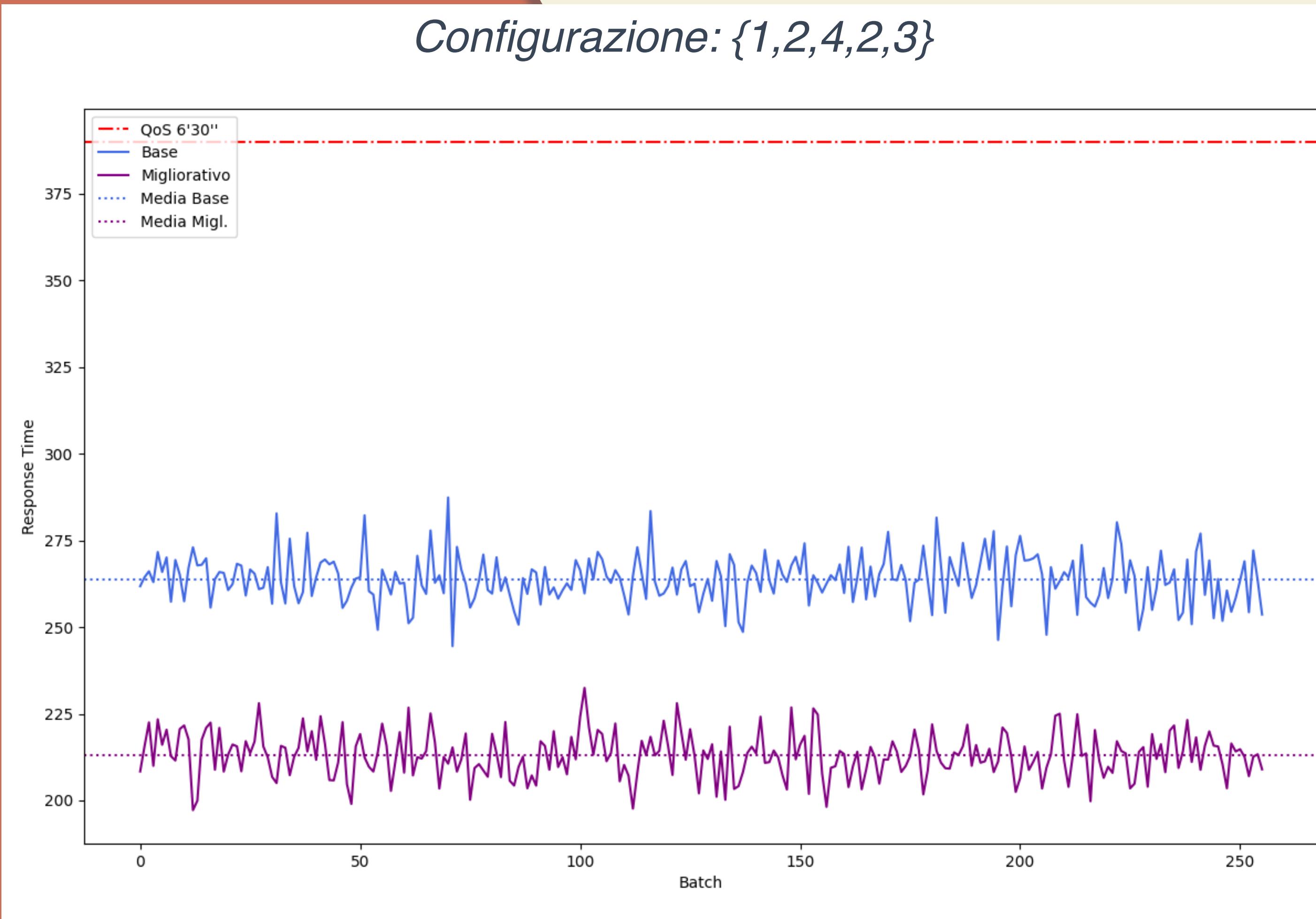
Minor numero di serventi totali del modello precedente

- ▶ Viene rispettato anche il QoS che richiede una percentuale minima di controllo del 25% da parte della polizia
- ▶ La media è compresa tra il 26.5% e il 27% ed è quindi il minimo numero di poliziotti per rispettare il QoS

MODELLO MIGLIORATIVO

Infinite Horizon Simulation – 3° Fascia

Configurazione: {1,2,4,2,3}



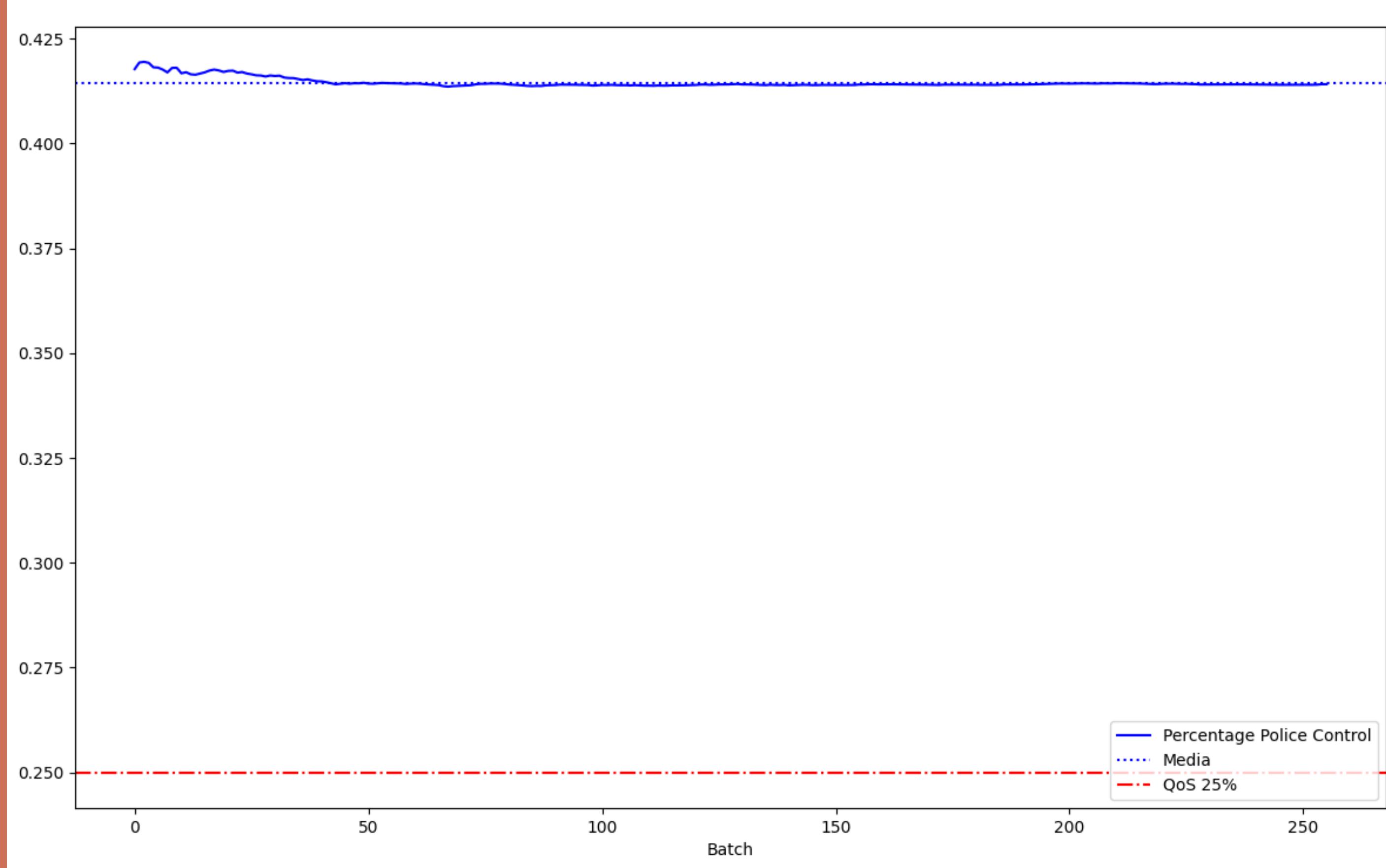
Stesso numero di serventi totali del modello precedente

- ▶ Il sistema è stazionario
- ▶ Risultato Analitico (da ‘queue_time.py’): 212.73469
- ▶ Risultato Sperimentale*: 212.91763 ± 0.84196
- ▶ Rispetta il QoS ed è migliorato rispetto al modello base mediamente di circa 51 secondi

MODELLO MIGLIORATIVO

Infinite Horizon Simulation – 3° Fascia

Configurazione: {1,2,4,2,3}



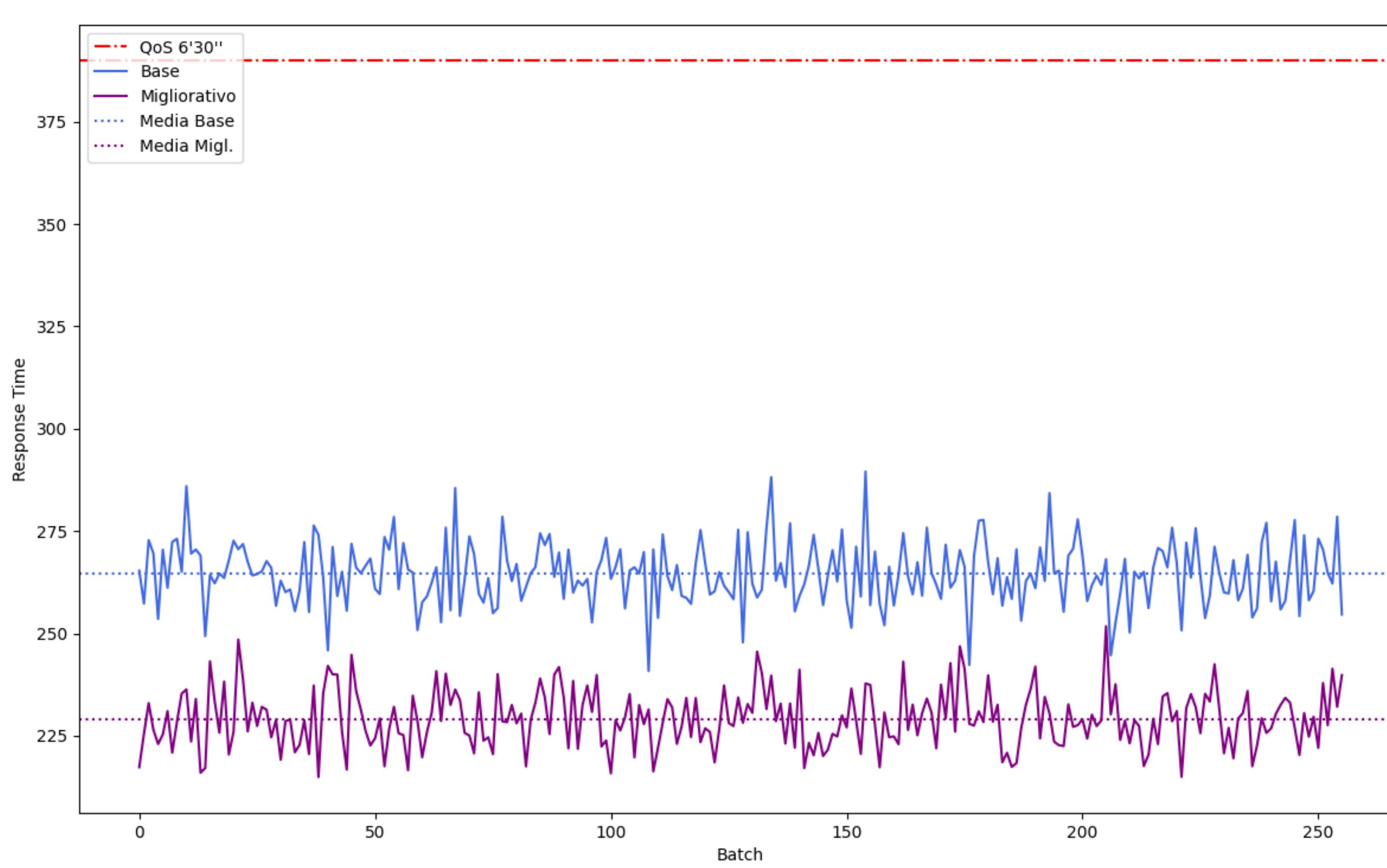
Stesso numero di serventi totali del modello precedente

- ▶ Viene rispettato anche il QoS che richiede una percentuale minima di controllo del 25% da parte della polizia
- ▶ La media è compresa tra il 40% e il 42.5% ed è quindi il minimo numero di poliziotti per rispettare il QoS e non variare il modello

MODELLO MIGLIORATIVO

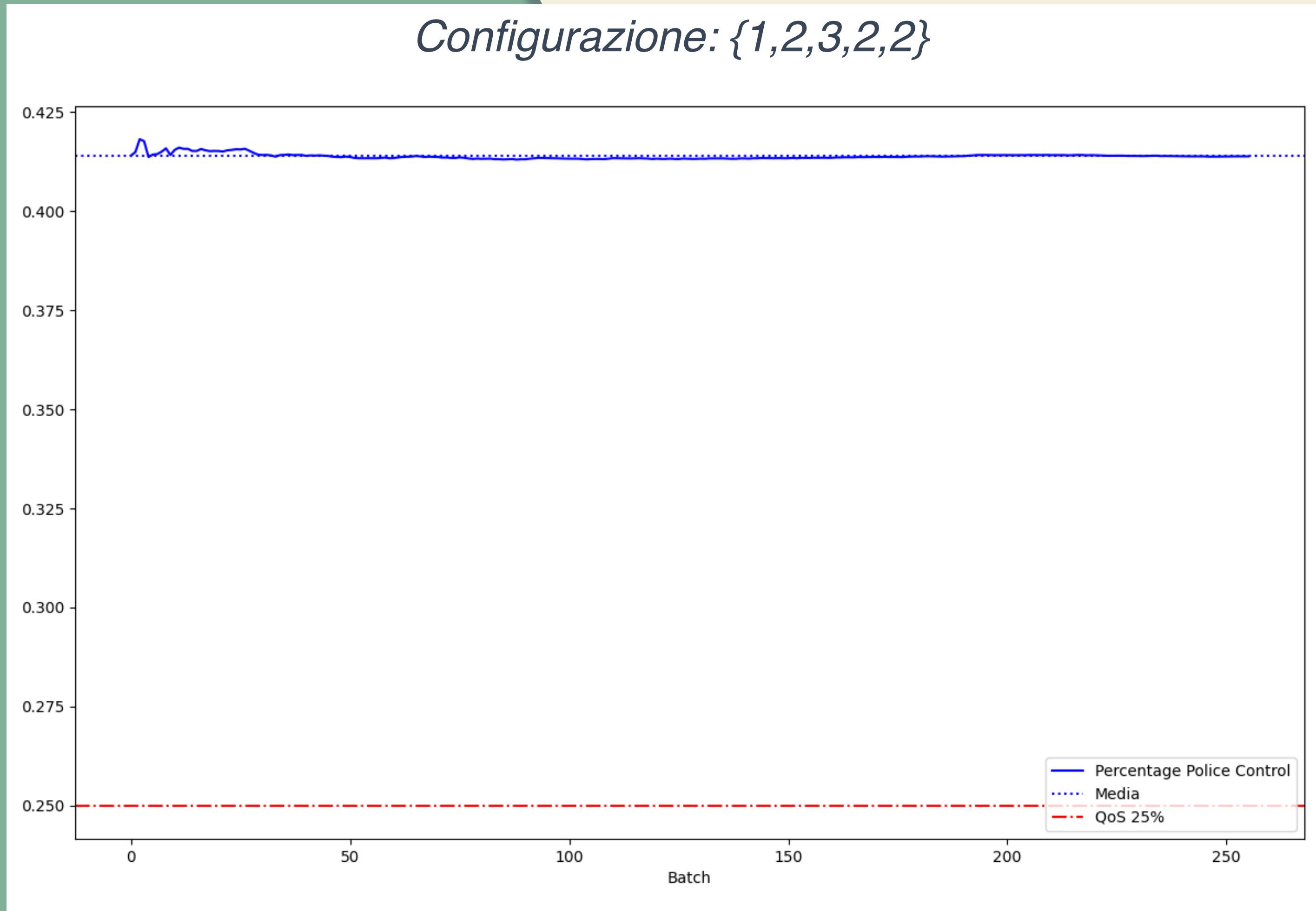
Infinite Horizon Simulation – 3° Fascia

Configurazione: {1,2,3,2,2}



Minor numero di serventi totali del modello precedente

- ▶ Il sistema è **stazionario**
- ▶ Risultato Analitico (da ‘*queue_time.py*’): **229.558**
- ▶ Risultato Sperimentale*: **229.13468 ± 0.86734**
- ▶ Rispetta il **QoS** ed è **migliorato** rispetto al modello base mediamente **di circa 35 secondi**



Minor numero di serventi totali del modello precedente

- ▶ Viene rispettato anche il **QoS** che richiede una percentuale minima di controllo del **25%** da parte della polizia
- ▶ La media è compresa tra il **40%** e il **42.5%** ed è quindi il **minimo numero di poliziotti** per rispettare il QoS e non variare il modello

Come nel modello precedente, la simulazione ad **orizzonte finito** è stata effettuata sulle **3 fasce orarie**.

I risultati sono stati mediati su **256 ripetizioni**:

- ▶ ogni ripetizione fornisce statistiche rappresentanti lo **studio del transiente** del sistema
- ▶ l'insieme delle statistiche rappresenta un punto del campione
- ▶ il sistema è **vuoto** ad inizio e fine simulazione

Il sistema viene assunto **dinamico** al cambio fascia:

- ▶ il **numero di serventi** nei centri viene aumentato e/o diminuito
- ▶ il **flusso di arrivo** varia come indicato nel modello delle specifiche

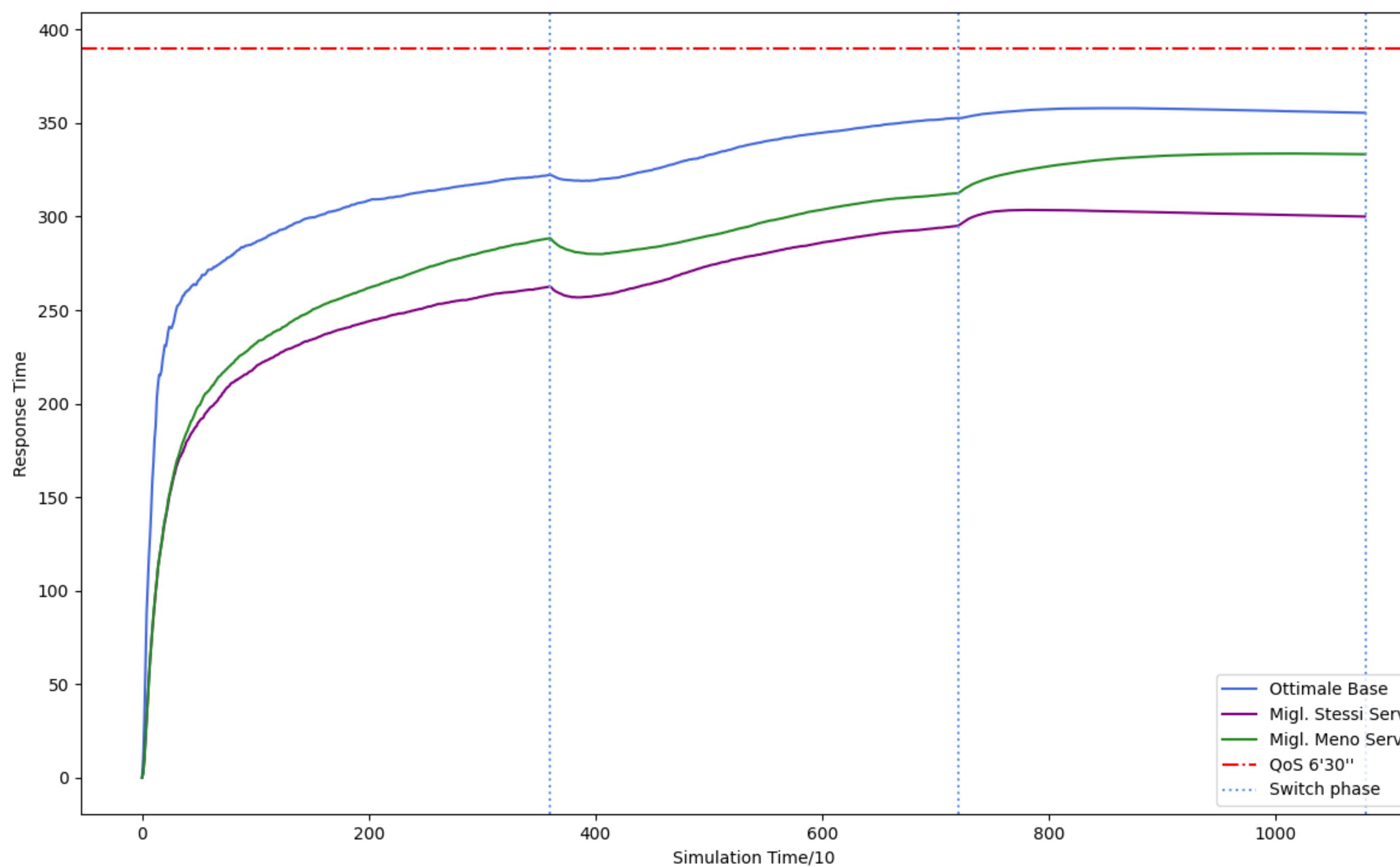
Per ogni ripetizione, sono state effettuate le misurazioni del **tempo di risposta del sistema ogni 10 secondi** (come somma dei tempi di risposta dei centri), per un totale di **1080 misurazioni per ripetizione**.

Gli **obiettivi** principali della simulazione ad orizzonte finito sono:

- ▶ analizzare il comportamento del sistema al cambio fascia
- ▶ verificare che le configurazioni individuate siano valide
- ▶ individuare la configurazione migliore per rispettare i QoS e minimizzare i costi
- ▶ analizzare il comportamento del sistema nel transiente

Configurazioni:

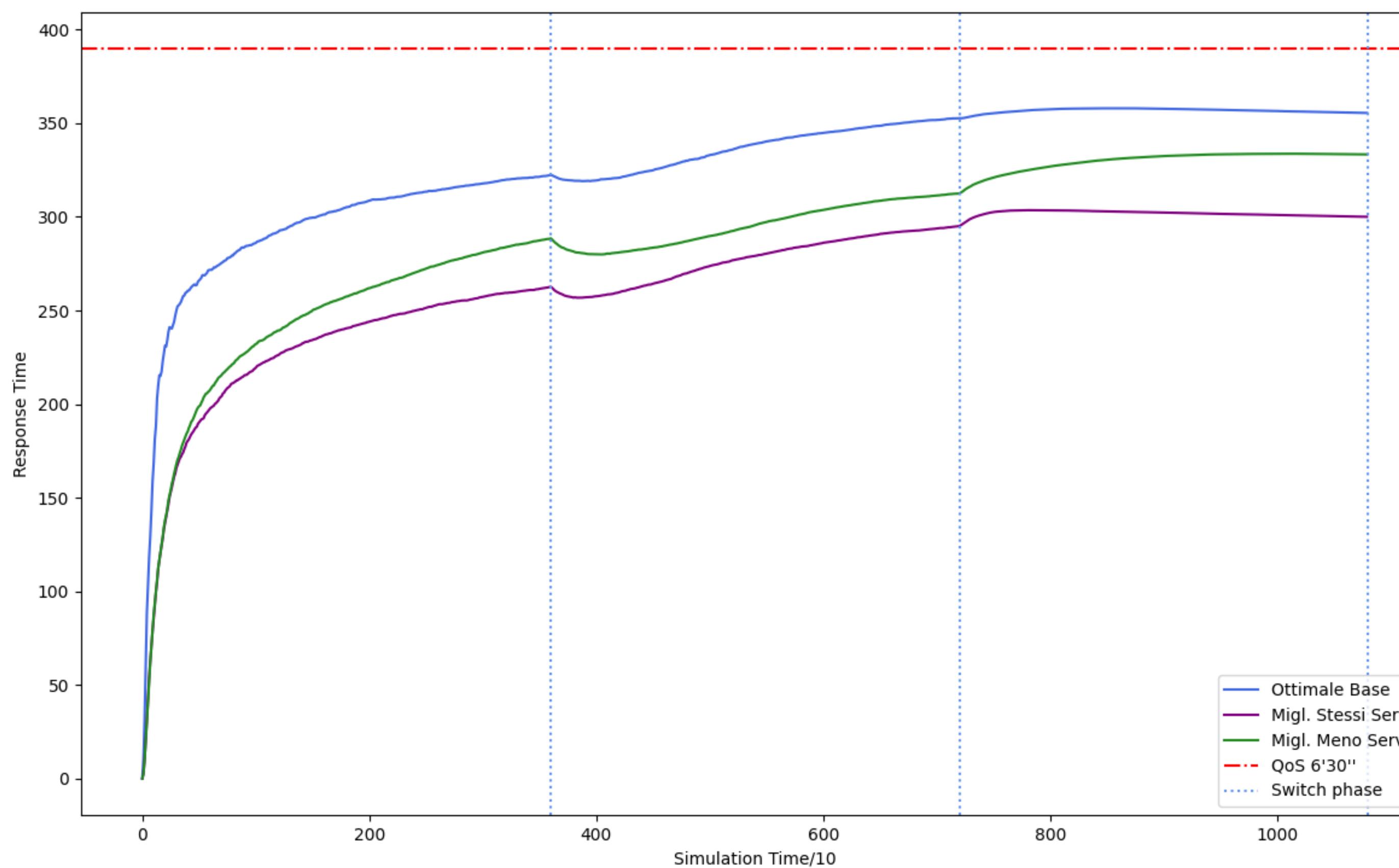
$\{1,5,14,6,10\} - \{1,11,27,13,20\} - \{1,2,4,2,3\}$
 $\{1,5,12,6,9\} - \{1,11,26,13,18\} - \{1,2,3,2,2\}$



- ▶ Entrambe le configurazioni nel modello migliorativo **rispettano il QoS** del tempo massimo di risposta del sistema
- ▶ Il modello migliorativo con stesso numero di serventi rispetto al precedente (**Base**) è quello con tempo di risposta totale minore

Configurazioni:

$\{1,5,14,6,10\} - \{1,11,27,13,20\} - \{1,2,4,2,3\}$
 $\{1,5,12,6,9\} - \{1,11,26,13,18\} - \{1,2,3,2,2\}$



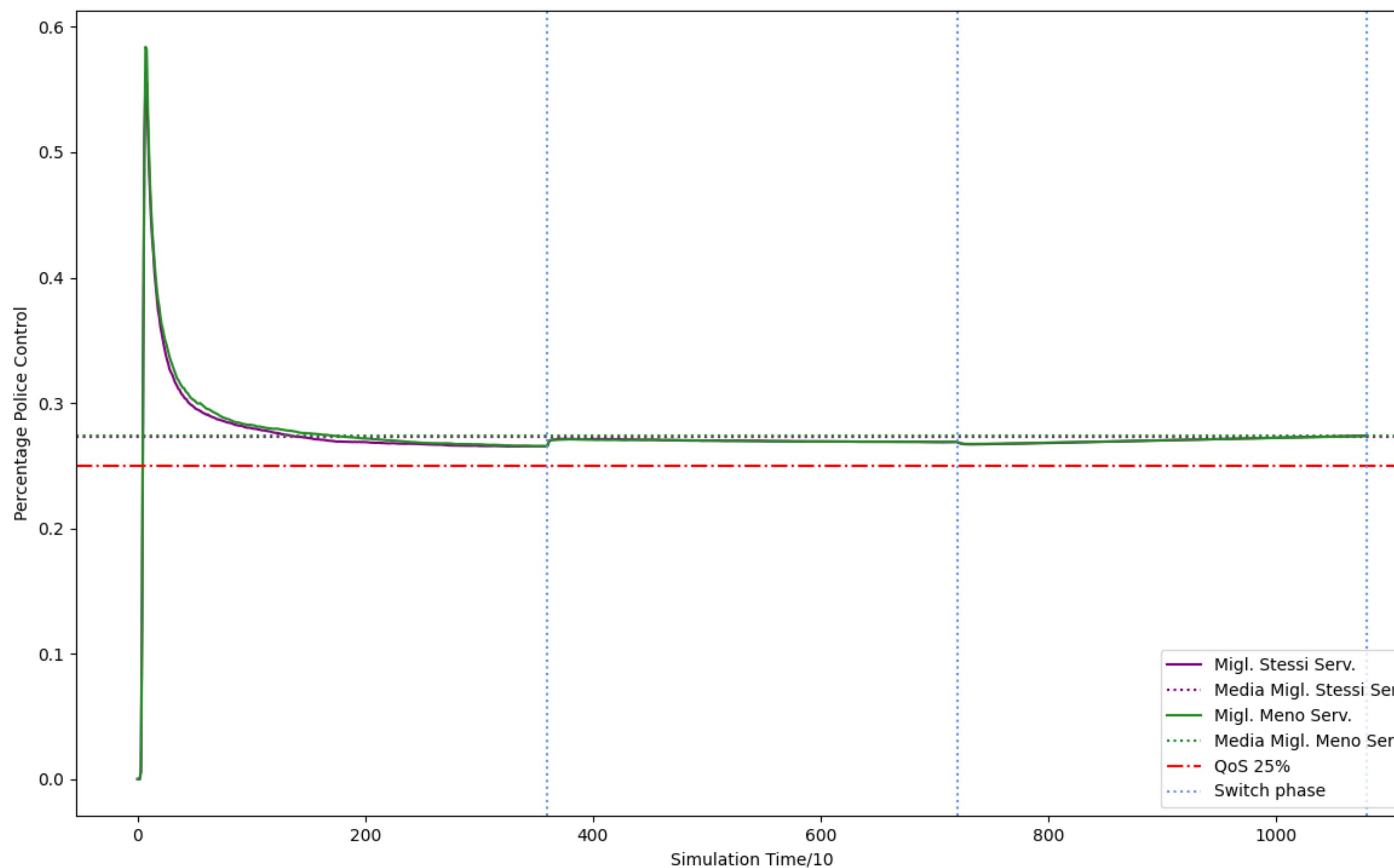
- ▶ Il modello migliorativo con minor numero di serventi rispetto al precedente (**Base**) è quello con tempo di risposta totale compreso tra gli altri due e con un risparmio di **100€**
- ▶ Risulta essere quindi un ***trade-off*** tra **prestazioni** e **prezzo**

MODELLO MIGLIORATIVO

Finite Horizon Simulation

Configurazioni:

$\{1,5,14,6,10\} - \{1,11,27,13,20\} - \{1,2,4,2,3\}$
 $\{1,5,12,6,9\} - \{1,11,26,13,18\} - \{1,2,3,2,2\}$



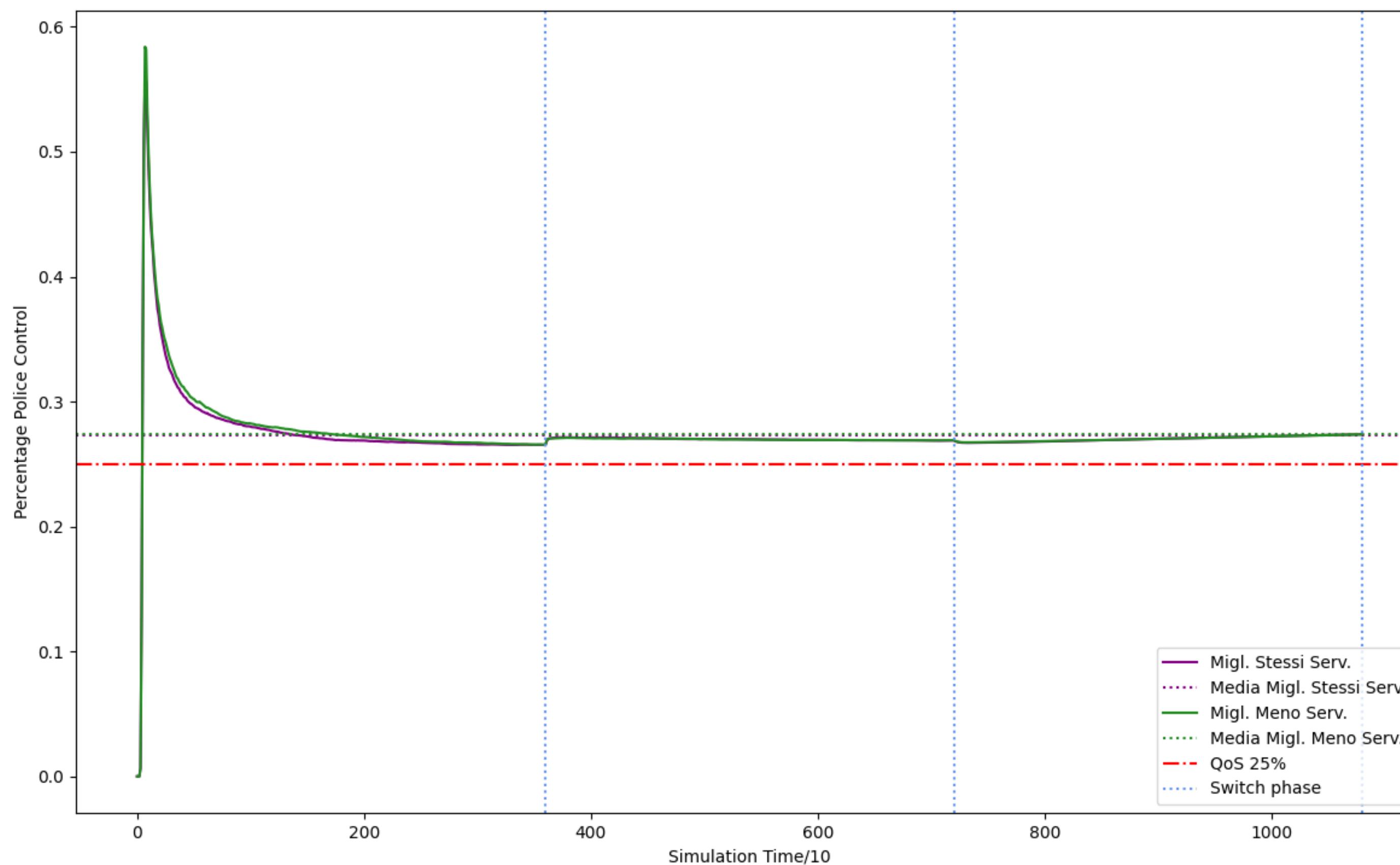
- ▶ Viene **rispettato anche il QoS** che richiede una percentuale minima di controllo del **25%** da parte della polizia con entrambe le configurazioni

MODELLO MIGLIORATIVO

Finite Horizon Simulation

Configurazioni:

$\{1,5,14,6,10\} - \{1,11,27,13,20\} - \{1,2,4,2,3\}$
 $\{1,5,12,6,9\} - \{1,11,26,13,18\} - \{1,2,3,2,2\}$



- ▶ Risulta interessante notare che in prima fascia la percentuale di controlli da parte della polizia è maggiore nella configurazione con meno serventi
- ▶ Come già visto in fase di validazione nel modello base, diminuendo i serventi in ‘Doc & Pat’ la percentuale di controllo aumenta

CONCLUSIONI

In entrambi i modelli presentati **vengono rispettati i QoS prestabiliti.**

Il **modello base** corrisponde al caso reale in cui i disabili hanno percorsi completamente separati. Nel **modello migliorativo**, tutti gli utenti attraversano gli stessi centri ma i disabili hanno code prioritarie e percorsi dedicati.

Il **modello migliorativo** proposto con lo **stesso numero di serventi**, e quindi con le stesse spese, permette di avere una notevole riduzione dei tempi di risposta.

Il **modello migliorativo** proposto con **minor numero di serventi**, permette di avere anch'esso una riduzione dei tempi di risposta rispetto al modello base ma prevede anche un **risparmio di 100€.**