

# Progetto di Simulazione di Sistemi

Samuele Evangelisti  
a.a. 2019/2020



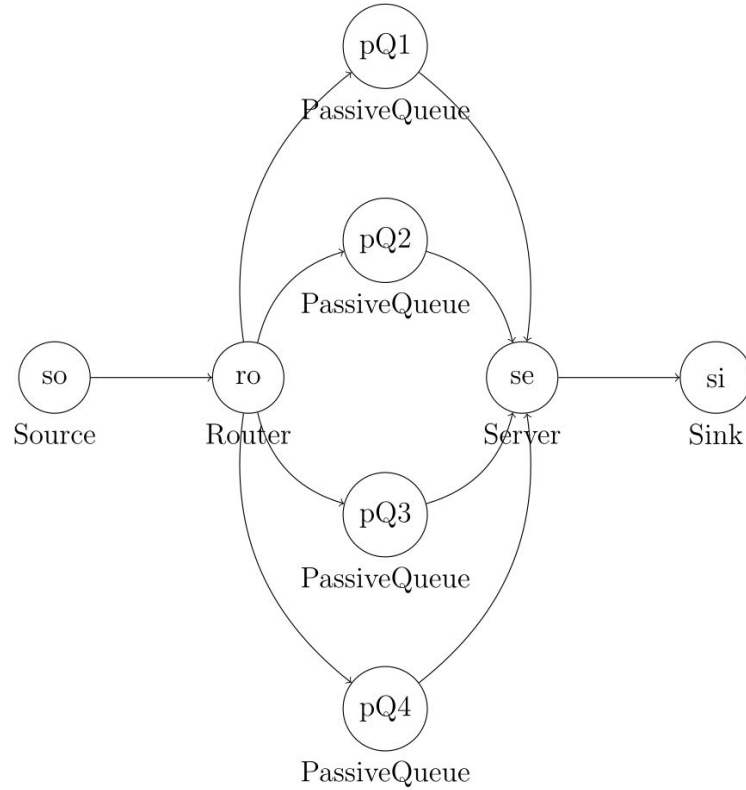
# Contenuti

1. Introduzione
2. Modello
3. *pssqueueinglib*
4. Misure di Prestazione
5. Risultati Sperimentali
6. Conclusioni
7. Riferimenti

# Introduzione

- Variante del modello riportato in [1] nella sezione 2
- Piattaforma: *OMNeT++ 5.6.1*
- *queueinglib* estesa e modificata in *pssqueueinglib*

# Modello (Rete)



# Modello (Rete)

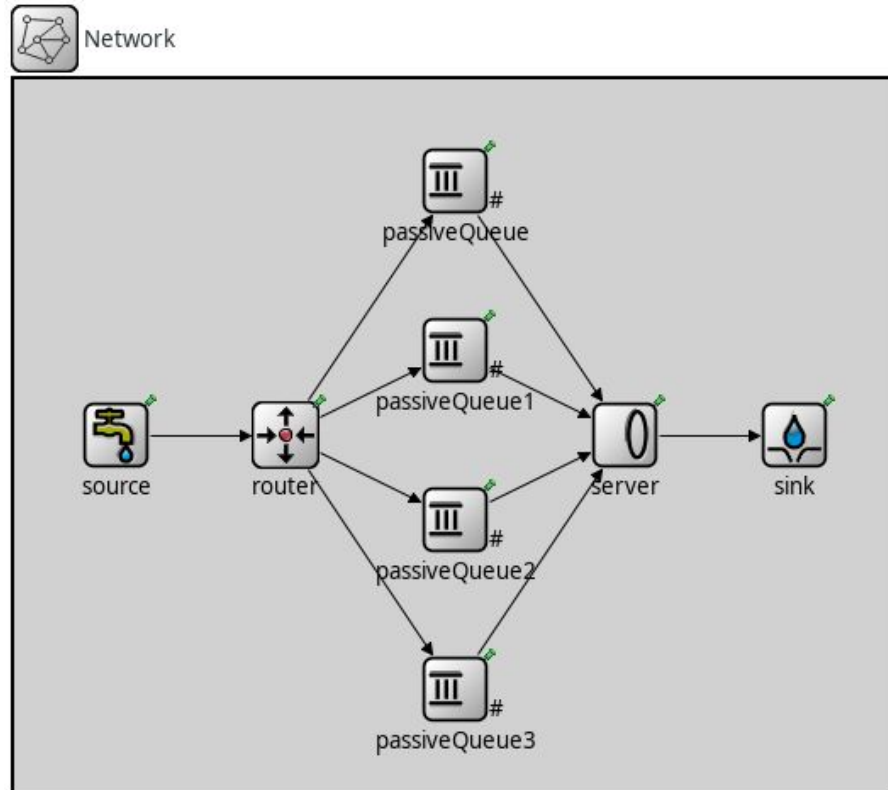
- **so**: esponenziale di media  $1/\lambda$
- **ro**: inoltra alle prime  $K$  *PassiveQueue*
- **pQ1**, ..., **pQ4**: capacità illimitata
- **se**: *exhaustive service*, polling circolare da **pQ1** a **pQ4**, esponenziale negativa di media  $1/\mu$
- **Job**: deadline di inizio servizio, distribuzione uniforme su  $[a, b]$

# Modello (Configurazioni)

- $\lambda$ : 2.0, 1.4, 1.2, 1.0
- K: 1, 2, 4
- $\mu$ : 3.0, 4.0
- [a, b]: [4.0, 6.0], [3.0, 7.0]

In totale sono presenti 48 configurazioni possibili

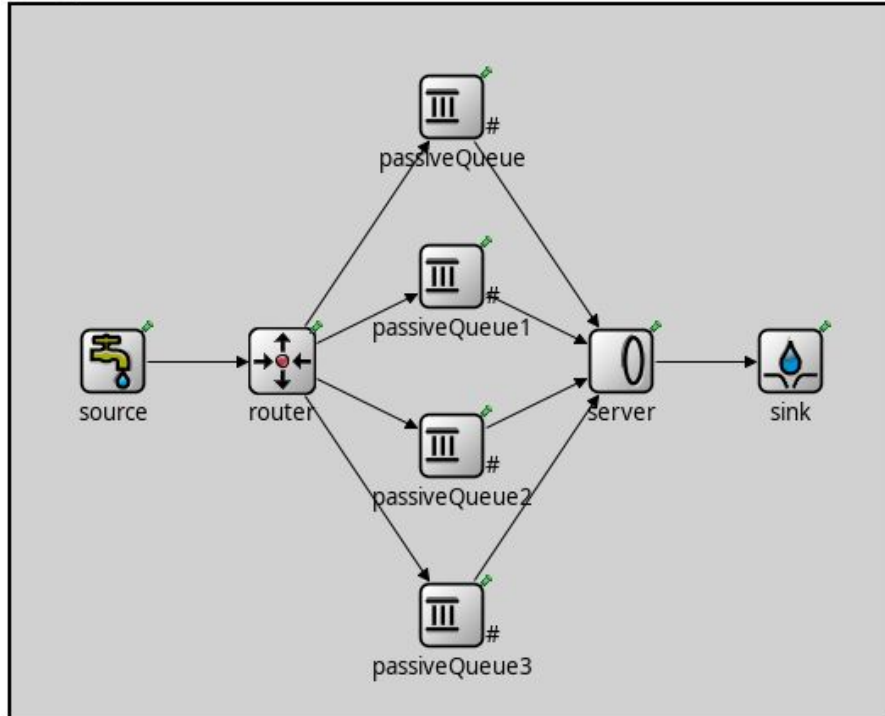
# Modello (OMNeT++)



# *pssqueueinglib (Job)*



Network



## Attributi:

- *simtime\_t absoluteDeadline*

## Metodi:

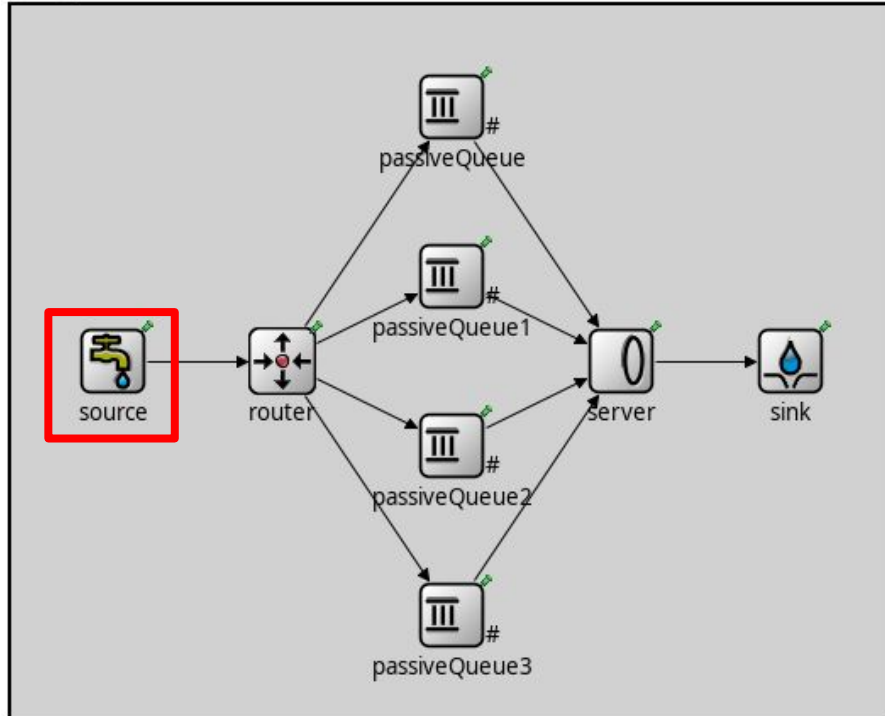
- *void setAbsoluteDeadline(simtime\_t absoluteDeadline) → Source*
- *simtime\_t getAbsoluteDeadline() → Server*



# *pssqueueinglib (Source)*



Network



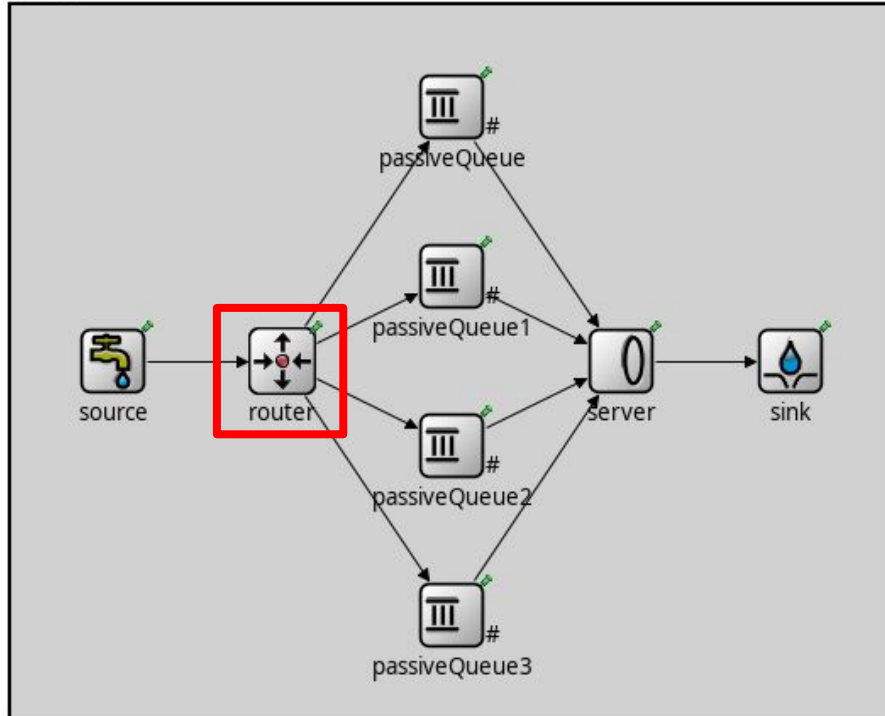
## Parametri:

- *double jobRelativeDeadline @unit(s) = default(0s) → Job*

# *pssqueueinglib (Router)*



Network



## Parametri:

- *int queueNumber = default(sizeof(out)-1)*

## Algoritmi di inoltra:

- *ALG\_PSSRANDOM*

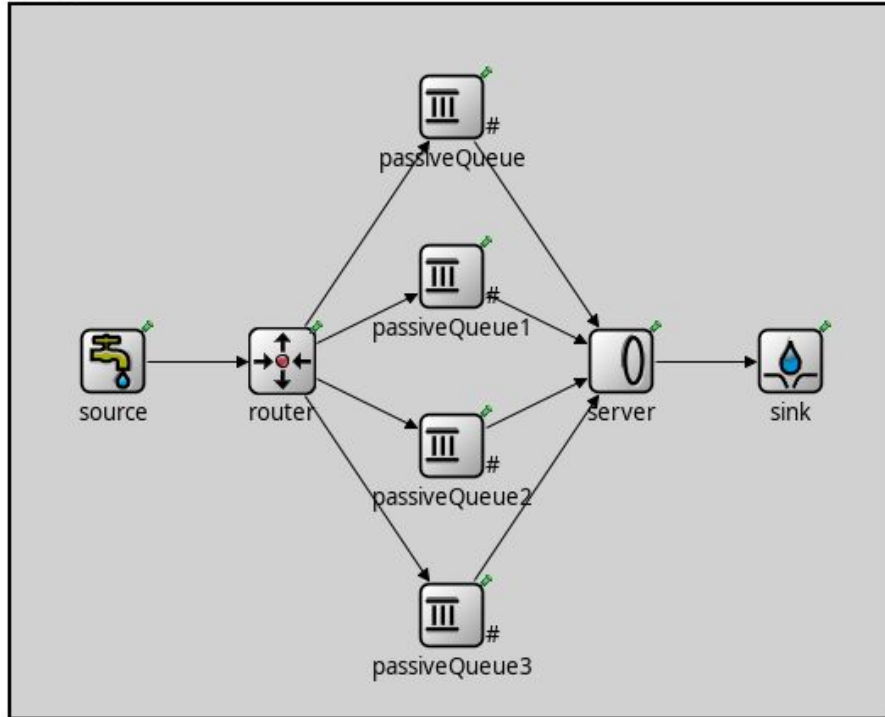
## Attributi

- *int queueNumber*

# *pssqueueinglib (SelectionStrategies)*



Network



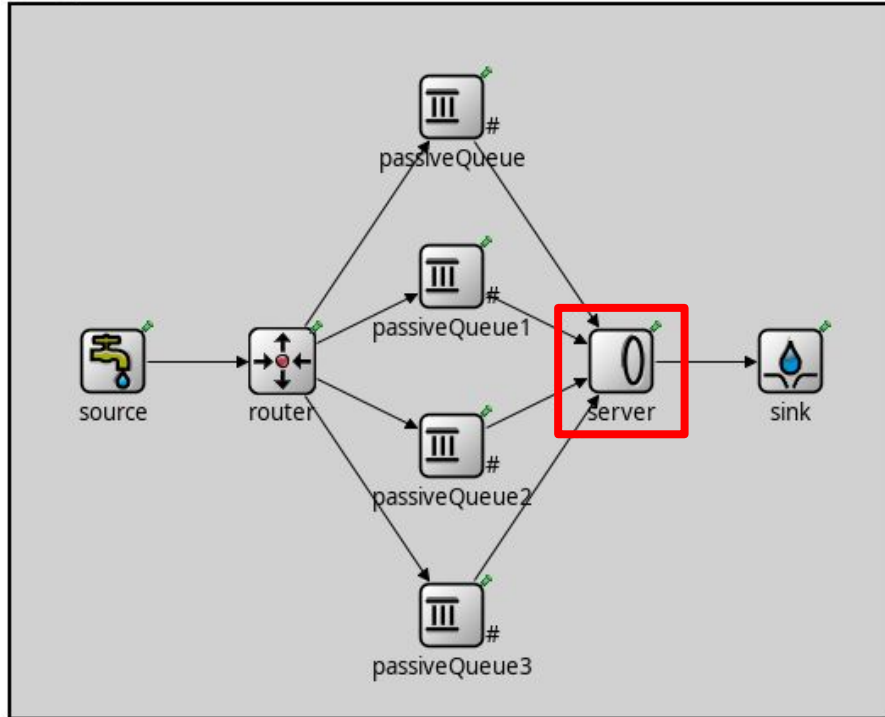
## Sottoclassi:

- Class `QUEUEING_API`  
*ExhaustiveServiceSelectionStrategy* : public  
*SelectionStrategy* → *Server*

# *pssqueueinglib (Server)*



Network



## Statistiche:

- `@signal[droppedForDeadline](type="long")`
- `@statistic[droppedForDeadline](title="drop event for deadline reached";record=vector?,count;interpolation mode=none)`

## Parametri:

- `bool checkJobDeadline = default(false)`

## Attributi:

- `simsignal_t droppedForDeadlineSignal`
- `bool checkJobDeadline`

# Misure di Prestazione

1. Mediana della distribuzione del tempo di risposta del sistema
2. Tempo minimo di permanenza dei *Job* nel sistema
3. Tempo massimo di permanenza dei *Job* nel sistema
4. Tempo medio di permanenza dei *Job* nel sistema
5. Numero medio di *Job* non serviti

Stima puntuale e intervallo di confidenza al 90% e al 95%.

Valori analizzati:

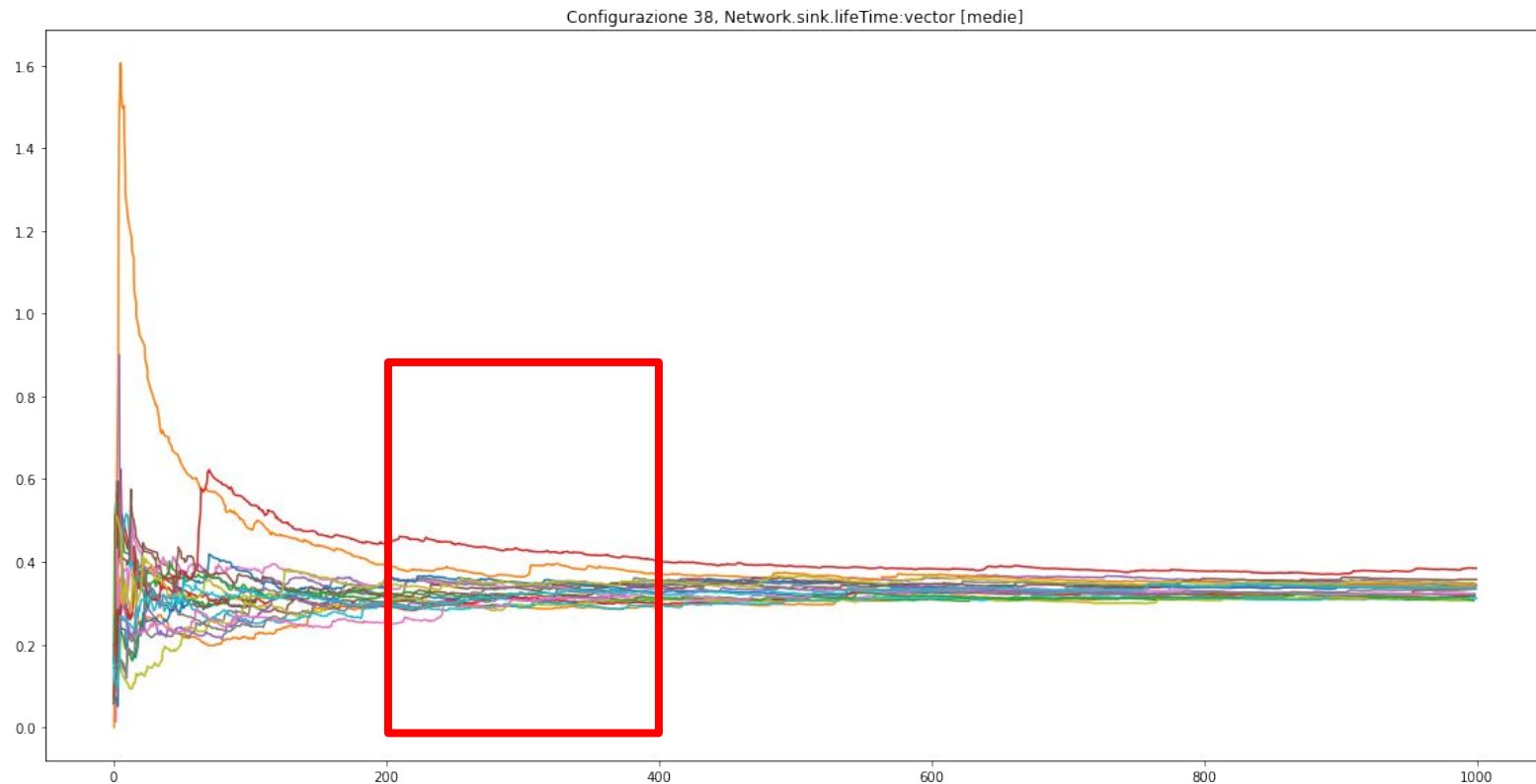
- *Network.sink.lifeTime:vector* (1, 2, 3, 4)
- *Network.server.droppedForDeadline:count* (5)

# Risultati Sperimentali

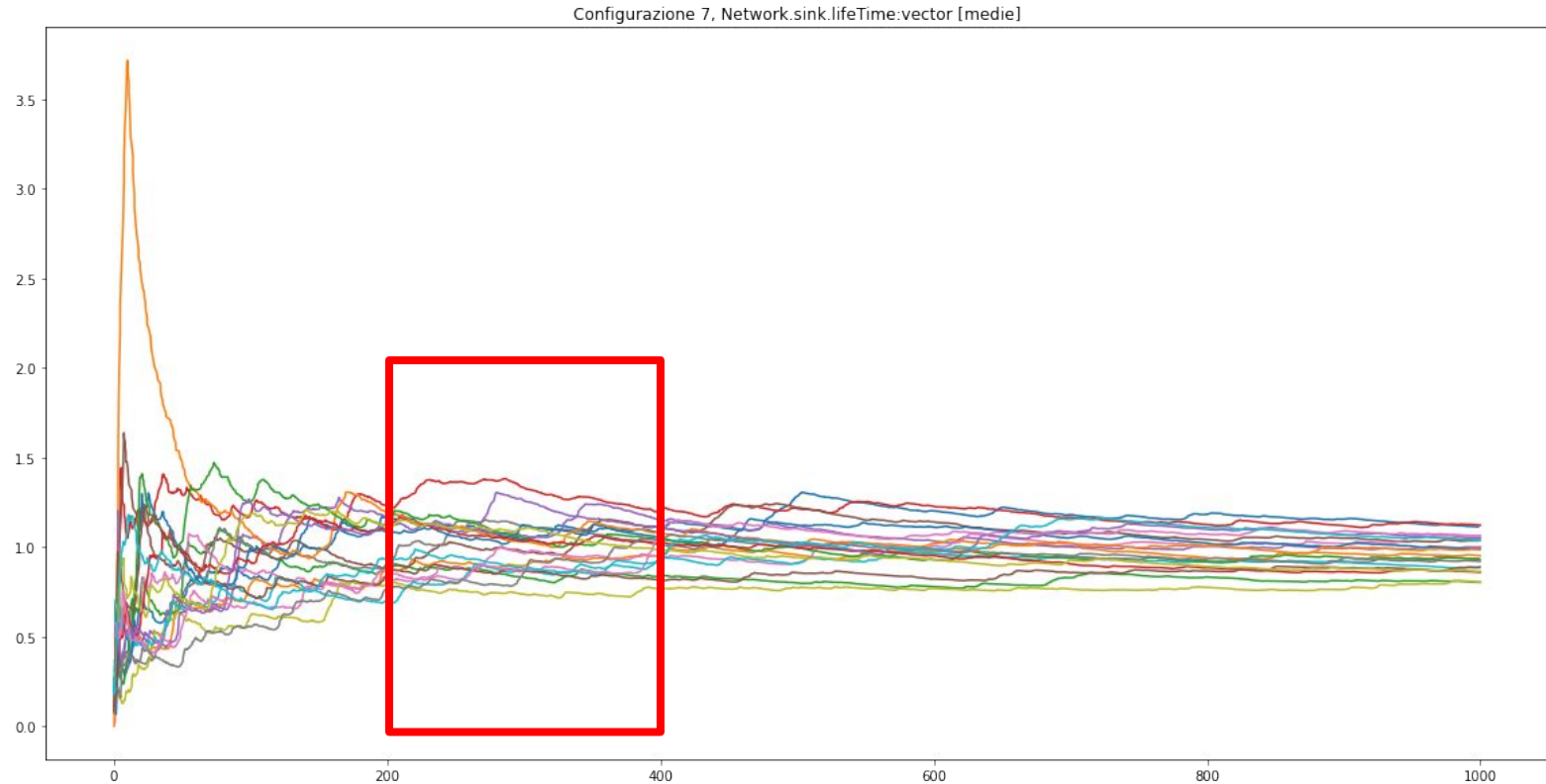
	$R_g(J)$	$R_t(J)$	
$J \geq 1000$	828	828	86.25%
$900 \leq J < 1000$	132	132	13.75%
$J < 900$	0	0	0.00%
	960		

Table 1: Analisi del numero di rilevazioni

# Risultati Sperimentali (Analisi del Transiente)



# Risultati Sperimentali (Analisi del Transiente)



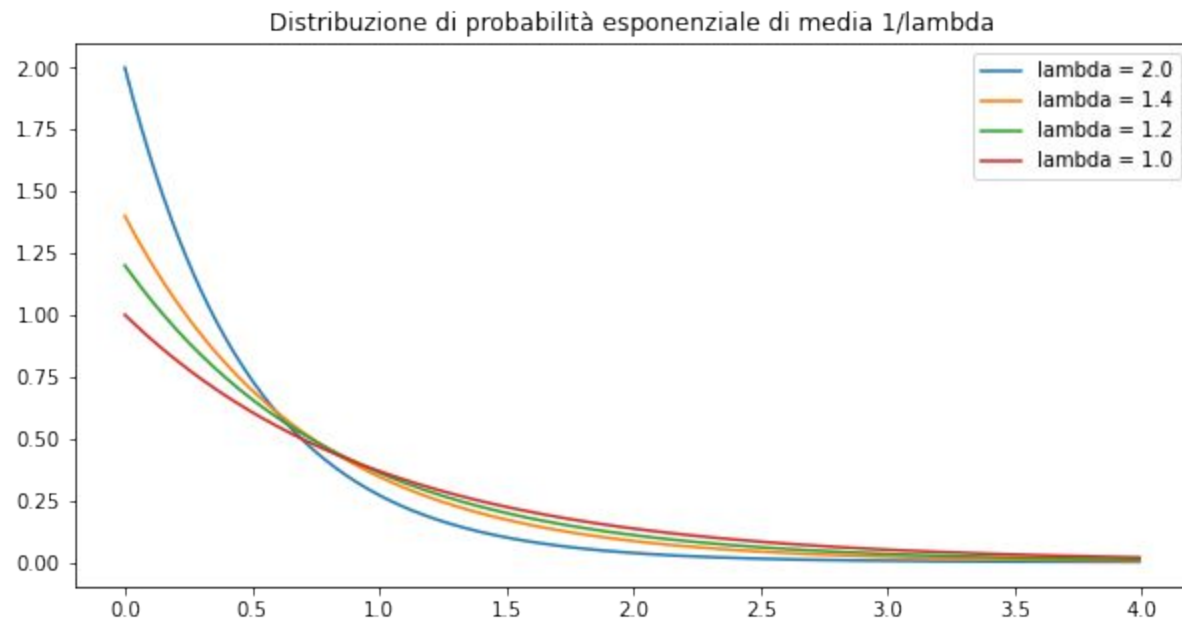


# Risultati Sperimentali (Analisi del Transiente)

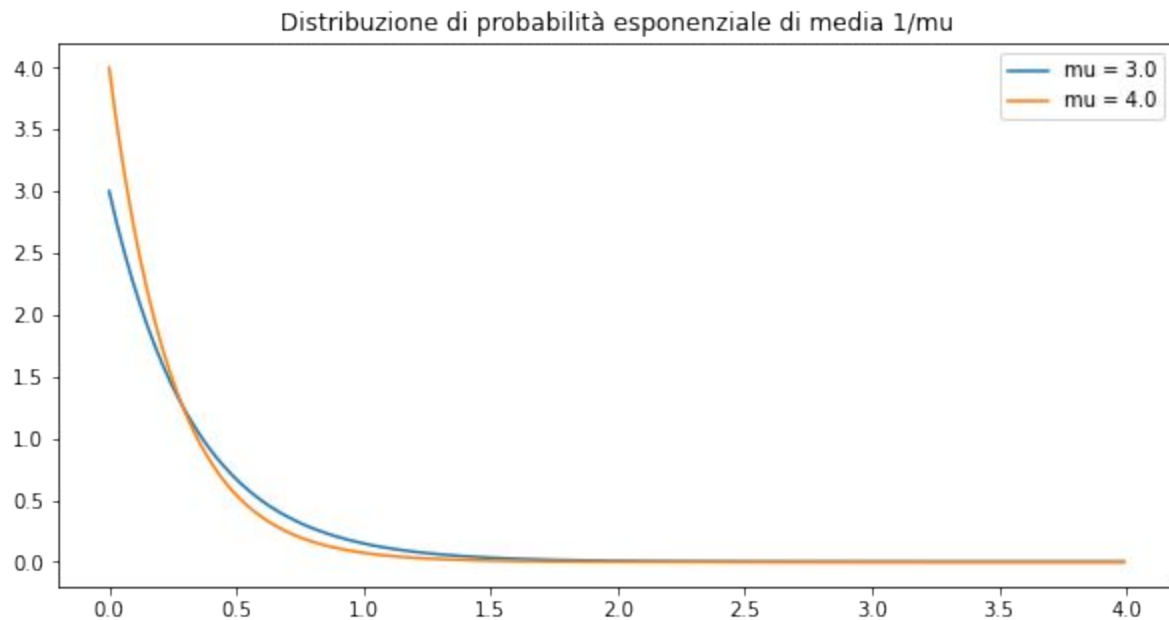
	$R_{t,s \geq 200}(J)$		$R_{t,s \geq 300}(J)$		$R_{t,s \geq 400}(J)$	
$J \geq 1000$	492	51.250%	262	27.291%	240	25.000%
$900 \leq J < 1000$	216	22.500%	218	22.708%	0	0.000%
$800 \leq J < 900$	114	11.875%	216	22.500%	216	22.500%
$700 \leq J < 800$	138	14.375%	150	15.625%	187	19.479%
$600 \leq J < 700$	0	0.000%	114	11.875%	197	20.520%
$500 \leq J < 600$	0	0.000%	0	0.000%	120	12.500%
$J < 500$	0	0.000%	0	0.000%	0	0.000%
	960					

Table 2: Analisi del numero di rilevazioni escludendo il transiente iniziale

# Conclusioni



# Conclusioni



# Conclusioni

Analizzando le distribuzioni di probabilità dei tempi di interarrivo e dei tempi di servizio ci si aspetta:

- Configurazione migliore:  $\lambda = 1.0$ ,  $\mu = 4.0$
- Configurazione peggiore:  $\lambda = 2.0$ ,  $\mu = 3.0$

# Conclusioni

$$\lambda = 1.0, \mu = 4.0$$

Le configurazioni che usano questi due parametri sono in media ottime e presentano un numero medio di *Job* non serviti circa nullo

$$\lambda = 2.0, \mu = 3.0$$

Sperimentalmente si nota un discreto numero di *Job* non serviti che cresce seguendo:

- $[a, b]: [4.0, 6.0] \rightarrow [3.0, 7.0]$
- $K: 1 \rightarrow 2 \rightarrow 4$

# Conclusioni (Tempo di Permanenza)

## Media:

- Configurazioni minime:  $\lambda = 1.0$ ,  $\mu = 4.0$ ,  $drop \approx 0$
- Configurazioni massime:  $\lambda = 2.0$ ,  $\mu = 3.0$ ,  $0 \lesssim drop \lesssim 3$

# Conclusioni (Tempo di Permanenza)

## Minimo

- Configurazioni minime:  $\lambda = 1.2$ ,  $\mu = 4.0$ ,  $drop \approx 0$
- Configurazioni massime:  $\lambda = 1.4$ ,  $\mu = 3.0$ ,  $0 \lesssim drop \lesssim 3$

## Massimo

- Configurazioni minime:  $K = 1$ ,  $\mu = 4.0$ ,  $drop \approx 0$
- Configurazioni massime:  $\lambda = 2.0$ ,  $\mu = 3.0$ ,  $0 \lesssim drop \lesssim 3$

# Riferimenti

1. [Sudipta Das, Debasis Sengupta and Lawrence Jenkins, Analysis of an M/M/1+G System Operated under the FCFS Policy with Exact Admission Control, 2012.](#)