

Performance Modeling of Computer Systems and Networks

Prof. Vittoria de Nitto Personè

Analytical models
(single resource)

Università degli studi di Roma Tor Vergata
Department of Civil Engineering and Computer Science Engineering

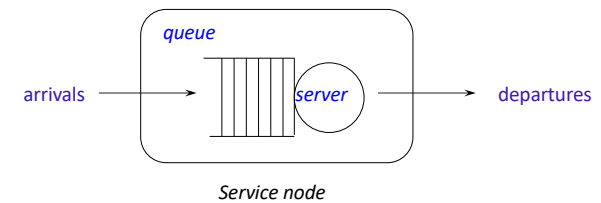
Copyright © Vittoria de Nitto Personè, 2021
<https://creativecommons.org/licenses/by-nc-nd/4.0/>



1

Discrete-event simulation
conceptual model

Single server queue



- *def. 1* a single server service node consists of a server plus its queue

Prof. Vittoria de Nitto Personè

2

2

Disciplina di scheduling: algoritmo utilizzato per selezionare il job che deve entrare in servizio.

Inizialmente assumiamo NON PRELAZIONE e servizio CONSERVATIVO:

"Non prelazione" significa che un job entrato in servizio non può essere interrotto, bensì devi completare l'esecuzione.

"Conservativo" vuol dire che, nel momento in cui ci sia qualche job in attesa del servizio e il server è libero, quest'ultimo deve subito iniziare a processare i job in attesa, e quindi non può rimanere "con le mani in mano".

Questo secondo aspetto potrebbe portare a degli "svantaggi" nel caso in cui conoscessimo i tempi di arrivo e le priorità dei job, infatti potremmo "aspettare" un job più importante di quelli attualmente in attesa.

Questo "svantaggio" è però annullato dal fatto che i job non sono interrompibili, quindi non possiamo ragionare in questa ottica.

Discrete-event simulation
conceptual model

def. 2 queue discipline (scheduling / service order):
the algorithm used when a job is selected
from the queue to enter service

FIFO – first in, first out
LIFO – last in, first out
random – serve in random order
Priority – typically shortest job first (SJF)

Prof. Vittoria de Nitto Personè 3

3

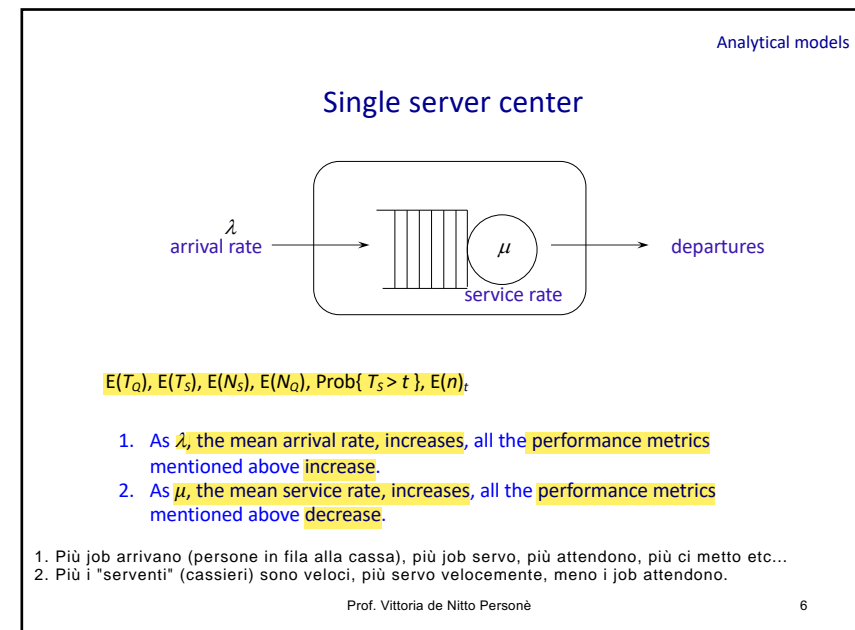
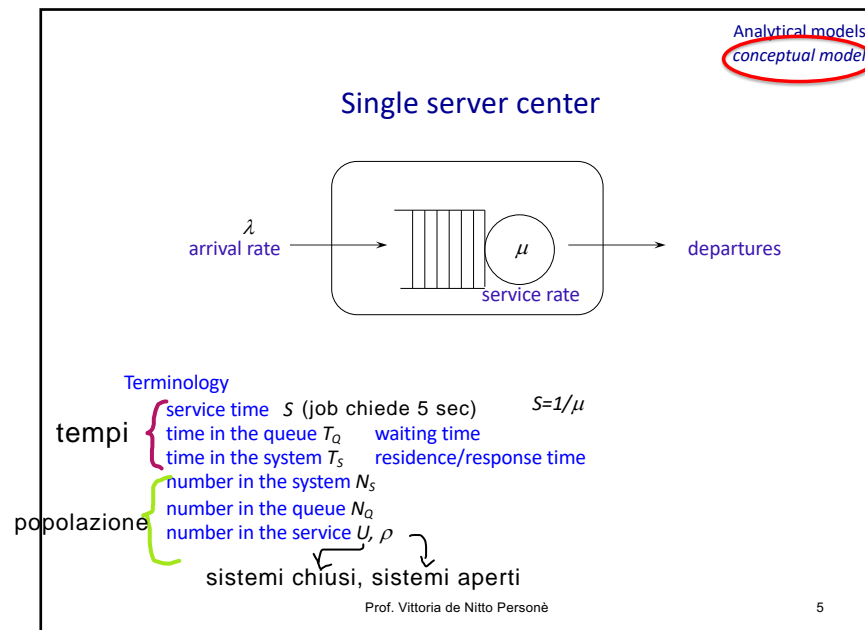
Discrete-event simulation
conceptual model

- FIFO (/ FCFS):
 - The order of arrival and departure are the same
 - A job cannot start service if the “previous” job has not left the node; this observation can be used to simplify the simulation
 - Unless otherwise specified, assume FIFO with infinite queue capacity
- service is *non-preemptive*
 - Once initiated, service of a job will continue until completion
- service is *conservative*
 - server will never remain idle if there is one or more jobs in the service node

Prof. Vittoria de Nitto Personè 4

4

Analizziamo una coda a servente singolo, possiamo individuare la seguente terminologia:



5

6

Spesso nei requisiti viene imposto che il tempo di risposta non deve superare un certo valore. Si può fare uno studio di probabilità sul tempo di risposta e studiare qual è la probabilità tale che T_s (tempo nel sistema) superi un certo valore " t ", ovvero $P(T_s > t)$, è un requisito di qualità.

Analytical models

Single server center

$E(T_Q), E(T_S), E(N_S), E(N_Q), \text{Prob}\{T_S > t\}, E(n)_t$

Def. throughput
 $t=1, E(n)_1$ n° of completions (departures) in the time unit
 (completamento nell'unità di tempo).
 Per un "t" generico, parlo del n° di completamenti nel tempo "t".

parlo di throughput per $t = 1$, NON TEMPO TOTALE.

Prof. Vittoria de Nitto Personè

7

7

Analytical models

Single server center

Def. utilization
 How can we "mathematically" define the utilization?

$\rho = \lambda / \mu$

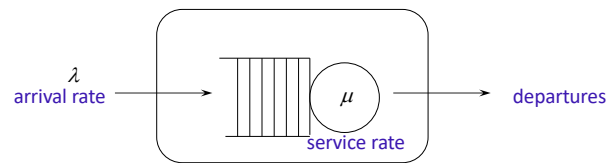
Prof. Vittoria de Nitto Personè

8

8 L'utilizzazione è la percentuale di tempo in cui si è occupati, rispetto al tempo che stiamo misurando.

Analytical models

Single server center


 $E(T_Q), E(T_S), E(N_S), E(N_Q), \text{Prob}\{T_S > t\}, E(n)_t$

$$E(T_S) = E(T_Q) + E(S)$$

ha senso perchè stiamo dicendo che la media del tempo passato nel sistema è uguale al tempo medio speso in coda + il tempo medio speso nel servente.

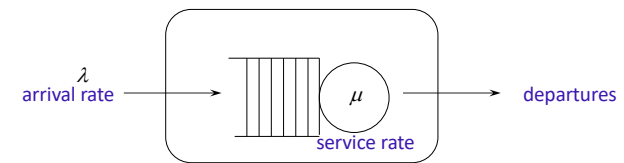
Prof. Vittoria de Nitto Personè

9

9

Analytical models

Single server center


 $E(T_Q), E(T_S), E(N_S), E(N_Q), \text{Prob}\{T_S > t\}, E(n)_t$

$$E(N_S) = E(N_Q) + E(\text{number in service})$$

è l'utilizzazione!
ρ

ha senso perchè stiamo dicendo che la popolazione media nel sistema è uguale alla popolazione media in coda + popolazione media nel servente.

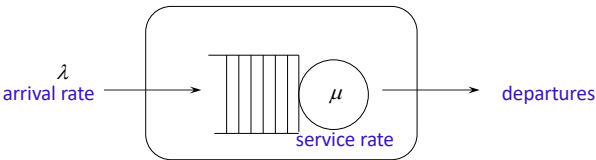
Prof. Vittoria de Nitto Personè

10

10

Analytical models

Single server center



$E(T_Q), E(T_S), E(N_S), E(N_Q), \text{Prob}\{T_S > t\}, E(n)_t$

$$E(N_s) = E(N_Q) + \rho$$

infatti la popolazione media nel servente non è nient'altro che l'utilizzazione media del servente.

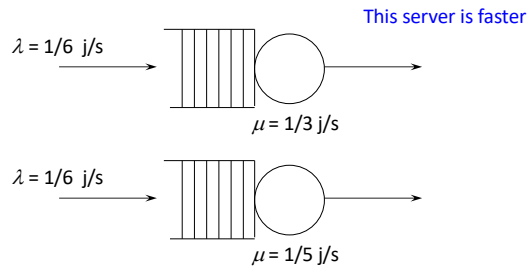
Prof. Vittoria de Nitto Personè

11

11

Analytical models

Single server center



This server is faster

Which system has greater throughput?

Notiamo che, in entrambi i casi, $\mu > \lambda$.
Se fosse stato il contrario? Arriverebbero più job di quelli che si possono servire, e quindi piano piano il centro si satura e genera coda infinita.

Prof. Vittoria de Nitto Personè

12

12

Entrambi i single server center hanno STESSO THROUGHPUT di 1/6 jobs/sec.
Ciò che cambia è il tempo di risposta e la lunghezza della coda, che nel server più veloce sono minori, ma ciò non ha nulla a che fare con il throughput.
TEMPI DI RISPOSTA SONO SCORRELATI DA THROUGHPUT.

Non importa quanto alto possa essere, il rate di completamento è comunque limitato, in questo esempio, dal rate di arrivo "rate in = rate out." Cambiare "mu" influenza il valore massimo del throughput, ma non quello attuale.

Analytical models

Single server center

$\lambda = 1/6$ j/s

$\mu = 1/3$ j/s

This server is faster

$\lambda = 1/6$ j/s

$\mu = 1/5$ j/s

$\lambda > \mu$

By assuming **job flow balance**, the **throughput is the same !!**

For both systems $X = \lambda = 1/6$ j/s

BUT the faster server shows the shorter queue and so shorter mean response time

In other words, improving the mean response time does not necessarily improve the throughput

Prof. Vittoria de Nitto Personè 13

Analytical models
basic laws

Single server center

(random)

If the center is in stochastic equilibrium (stationary condition),

$E(n)_i = X = \lambda$

Throughput is independent of the service rate μ throughput non dipende da mu

$\lambda < \mu, \rho = \lambda / \mu < 1$

If the center is NOT in stochastic equilibrium,

$E(n)_i = X = \mu$

the center cannot work off the arrival rate, the queue grows unlimited

coda cresce e va a infinito.

$\lambda > \mu,$

Prof. Vittoria de Nitto Personè 14

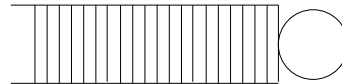
13

Concettualmente, possiamo pensarla così:
 Ho due cassieri: cassiere A serve una persona in 3 minuti ($\mu = 1/3$), cassiere B in 5 minuti ($\mu = 1/5$).
 Una persona si aggiunge ad una delle due code 10 minuti dopo la precedente. (quindi $\lambda = 1/10$)
 In questo caso non importa quanto il cassiere sia veloce, perché siamo comunque vincolati dagli arrivi dei clienti (quelli che arrivano = quelli che escono).

14

In pratica il throughput è il minimo tra λ e μ , perché vedere il minimo tra loro due mi dà informazioni anche sulla presenza o assenza di equilibrio stocastico.

Single server center

What's up if $\lambda > \mu$?

the center cannot work off the arrival rate, the queue grows unlimited

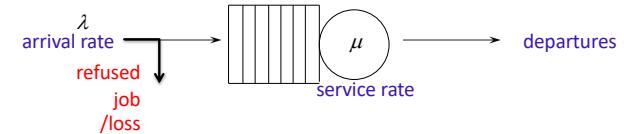
proporziono rispetto a T, in quanto
il periodo di osservazione influisce sulla
popolazione media nell'unità di tempo.

$$E(N_Q \text{ in } T) \geq \lambda T - \mu T = T(\lambda - \mu) \rightarrow \infty \text{ as } T \rightarrow \infty$$

media in coda nel tempo T

E' come dire:
nella coda ci sono 3 job/s
ovviamente se osservo per 2 secondi
ne avrò 6 job e così via!

15

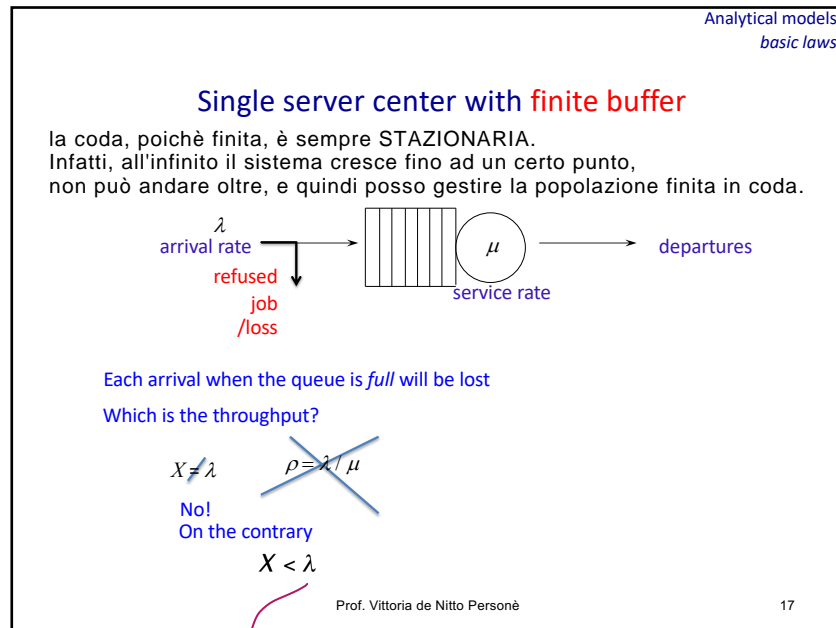
Single server center with **finite buffer**Each arrival when the queue is *full* will be lost

Which is the throughput?

Finora abbiamo assunto code INFINITE, coda accade con coda FINITA?

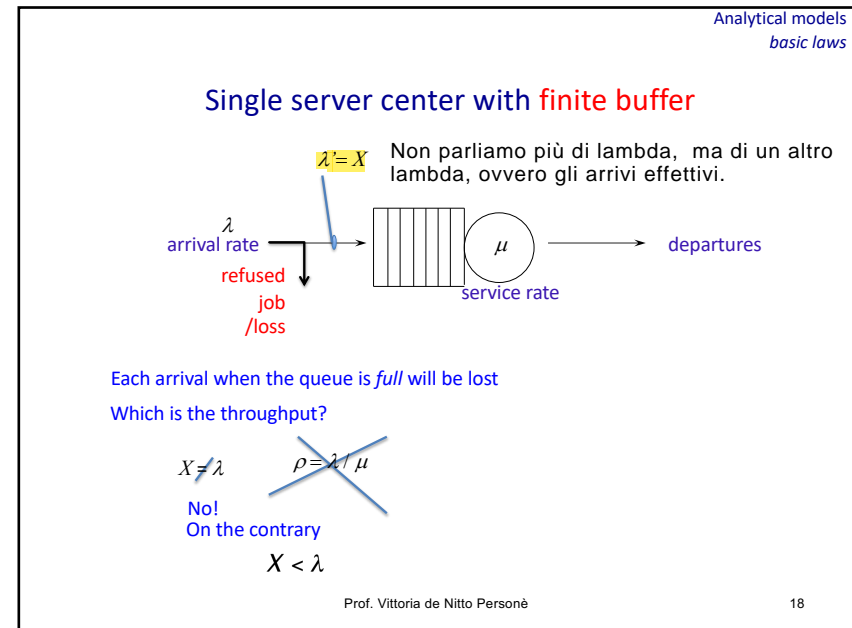
16

Innanzitutto dobbiamo mantenere un CONTATORE per il numero di jobs in coda, in quanto all'arrivo di un nuovo job in coda dobbiamo poter dire se tale job è aggregabile in coda, o se dobbiamo scartarlo. Inoltre dobbiamo mantenere anche degli indici per identificare il tasso di perdita.



17

Questo perchè ricevo meno di lambda, in quanto devo considerare la perdita del sistema. Cosa possiamo dire dell'utilizzazione? Possiamo solamente dire che non è semplicemente lambda/mu, per dire altro dobbiamo fare analisi stocastica.



18

Consider a web server with a mean processing rate of 1.2 job/s.
If the server receives requests with a rate of 0.45 job/s and it has 0.225
enqueued jobs on average, determine:

- a) the average utilization
- b) the average response time.

During rush hours the arrival rate grows of 20% and the average number of
enqueued jobs becomes 0.3681818.

Determine:

- c) the performance metrics a) and b)
- d) which further increasing in arrival rate makes the server collapsing
- e) the performance metrics a) and b) for the limiting case d).

Let us consider a server that processes jobs with rate 0.8 jobs/s.
By assuming that the server receives jobs with a rate depending on the time slot as
follows:

- 8.00 a.m. – 12.00 a.m. average arrival rate 1.5 jobs/s
- 12.00 a.m. – 2.00 p.m. average arrival rate 0.5 jobs/s
- 2.00 p.m. – 7.00 p.m. average arrival rate 1.5 jobs/s
- 7.00 p.m. – 9.00 p.m. average arrival rate 0.5 jobs/s
- 9.00 p.m. – 8.00 a.m. average arrival rate 0.05 jobs/s

Determine:

- a) average arrival rate per day (24 hours)
- b) average utilization per day
- c) average throughput per day
- d) average throughput for each time slot

Please, justify and comment the results by indicating the used laws.

Analytical models
basic laws

Multi Server Queue

server in // e indipendenti
(non collaborano per minimizzare
un singolo job più lungo)

(per ogni server)

$E(S_i)$ $E(S)$

Qui abbiamo la solita coda di arrivi, e diversi server omogenei in parallelo con frequenza di servizio μ , tale frequenza di servizio è per ogni server. Se un server è idle, allora la coda è vuota (perchè è conservativo). Se la coda NON è vuota (c'è qualcuno) questo vuol dire che tutti i server sono occupati (perchè sono conservativi).

Prof. Vittoria de Nitto Personè 21

21

Analytical models
basic laws

Multi Server Queue

$E(S_i) = 1 / \mu$ $E(S) = 1 / m\mu = E(S_i) / m$

tempo che passa da quando il job viene preso in carico dal server i-esimo a quando viene rilasciato dal server i-esimo

tempo che passa da quando UN job viene preso in carico da UN server generico a quando viene rilasciato da UN ALTRO server generico

Prof. Vittoria de Nitto Personè 22

22

Vado alle poste.
Ci sono due file. Mi concentro sulla fila1 e vedo ogni quanto tempo il dipendente della fila1 termina il servizio con il cliente.

Vado alle poste.
Ci sono due file. Non mi concentro più sulla fila singola, ma le osservo entrambe, e vedo generalmente ogni quanto un dipendente (di una delle due file, non mi importa quale) si libera.

Analytical models
basic laws

Multi Server Queue

$$E(N_s) = \begin{cases} E(N_q) + \rho & \text{if } m=1 \text{ servere singolo} \\ E(N_q) + m\rho & \text{if } m>1 \text{ servere multiplo} \end{cases}$$

But how is the utilization defined for the multiserver case?

Prof. Vittoria de Nitto Personè 23

23

Analytical models
basic laws

Multi Server Queue

$$\rho_i = \frac{\lambda_i}{\mu} = \frac{\lambda}{m\mu}$$

$$\rho_{glob} = \frac{\lambda}{\mu_{glob}} = \frac{\lambda}{m\mu} \quad \text{tasso medio in entrata} \quad \text{tasso max in uscita}$$

Idealmente, ogni server riceverà una porzione di arrivi rapportata a quanti server ci sono! (al supermercato, se ho due cassieri, le file saranno più o meno eque).

Nell'utilizzo globale, ho m serveri che lavorano, quindi $m \cdot (\mu)$

Prof. Vittoria de Nitto Personè 24

24

L'utilizzazione del singolo servente, e globale hanno stessa "formula", ma non ci danno le stesse informazioni!

L'utilizzazione del singolo servente mi dice mediamente QUANTO viene utilizzato quel singolo servente. (es: ogni server è sfruttato al 65%)

L'utilizzazione globale mi dice mediamente IL NUMERO DI SERVER UTILIZZATI su m totali (es: mediamente, uso 3 server su 5)

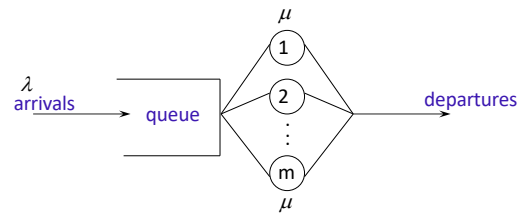
Ad esempio, se $m = 10$, e utilizzazione = 0.5

- Per il singolo servente: è sfruttato al 50% mediamente.
- A livello globale, utilizzo $10 \cdot 0.5 = 5$ server mediamente.

Ovviamente, se la popolazione nel sistema cresce e va a infinito, utilizzo tutti i server presenti!

Analytical models
basic laws

Multi Server Queue



$$\rho = \begin{cases} \frac{\lambda}{\mu} = \lambda E(S_1) & \text{if } m = 1 \\ \frac{\lambda}{m\mu} = \frac{\lambda E(S_1)}{m} & \text{if } m > 1 \end{cases}$$

• sono inversi tra loro!
service rate = 1/ tempo di servizio

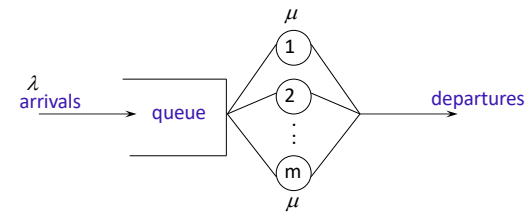
Prof. Vittoria de Nitto Personè

25

25

Analytical models
basic laws

Multi Server Queue



$$\rho_i = \rho_{glob} = \frac{\lambda}{m\mu}$$

Prof. Vittoria de Nitto Personè

26

26

Analytical models
basic laws

Multi Server Queue

λ arrivals → queue → $\begin{matrix} \mu \\ 1 \\ 2 \\ \vdots \\ m \\ \mu \end{matrix}$ → departures

tempo risposta = $E(T_s) = E(T_q) + E(S_i)$

tempo di coda (prima di ricevere il servizio) → $E(T_q)$
 completamento di uno specifico job → $E(S_i)$
NON BISOGNA CONFONDERSI CON E(S)

Poichè E(S) è il tempo medio in cui uno degli m server si libera.
 Ovvero da quando uno qualsiasi entra a quando uno qualsiasi esce (diverso da chi è entrato) cioè dopo quanto tempo se ne libera uno a caso tra queglii "m", in media!
 E(S_i) è di un singolo server, da quando un job entra a quando quel job esce

Prof. Vittoria de Nitto Personè

27

- 27 Facciamo un parallelismo con la fila al supermercato:
 Il tempo di risposta (esco dalla fila dopo aver pagato)
 è dato da quanto tempo aspetto in coda + il tempo in cui il cassiere scansiona prodotti e mi fa pagare.
 Questo secondo tempo è E(S_i).
 E(S) è quanto mediamente uno dei vari cassieri si libera, non quanto ci mette a servirmi.

Analytical models
basic laws

Multi Server Queue

λ arrivals → queue → $\begin{matrix} \mu \\ 1 \\ 2 \\ \vdots \\ m \\ \mu \end{matrix}$ → departures

$E(T_s) = E(T_q) + E(S_i)$
 $E(T_q) = f(\lambda, m, E(S_i))$

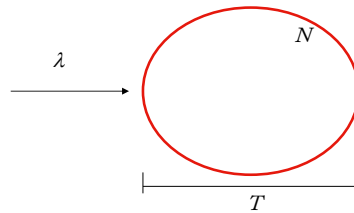
Dipende anche da "m" perchè invece di usare E(S_i) sfrutto E(S) globale, in quanto tutti i serventi sono uguali, e per ottenere E(S_i) faccio $E(S_i) = E(S) * m$

Prof. Vittoria de Nitto Personè

28

28

Little's law is very important for its broad applicability.
In general, we can see Little's law as applied at a black box:
it states relations between mean values



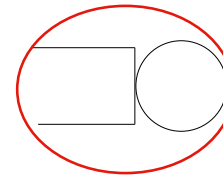
Little's Law (1961)

- (a) queue discipline is FIFO,
- (b) service node capacity is infinite,
- (c) flow balance

$$N = \lambda T$$

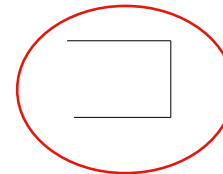
If λ is the mean arrival rate, T is the mean residence time in the black box, N is the mean population in the black box, the theorem asserts that, if the system is "stable", the mean population is given by the "mean arrival flow" multiplied the mean time the jobs spend in the black box

29



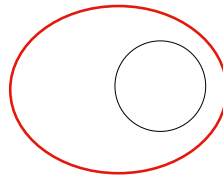
If the black box is the whole center, the theorem is applied to the center mean population:

$$E(N_s) = \lambda E(T_s)$$



If the black box is just the queue, the theorem is applied to the queue mean population:

$$E(N_q) = \lambda E(T_q)$$



If the black box is just the server, the theorem is applied to the server "mean population", in other words to the utilization:

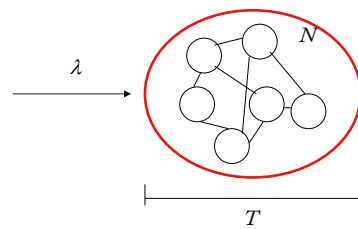
$$\rho = \lambda E(S)$$

Prof. Vittoria de Nitto Personè

30

30

But if the black box is a network of centers, anyway interconnected,



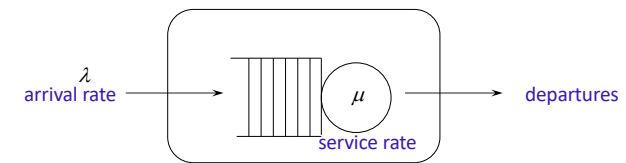
$$N = \lambda T$$

The theorem is applied to the entire network!!

31

Analytical models

Single server center



$$\begin{array}{ll}
 E(T_s) = E(T_q) + E(S) & \text{Little's law} \\
 E(N_s) = \lambda E(T_s) & \\
 E(N_s) = E(N_q) + \rho & \\
 E(N_q) = \lambda E(T_q) &
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{l}
 E(T_s) = \frac{E(N_s)}{\lambda} \\
 E(T_q) = \frac{E(N_q)}{\lambda}
 \end{array}$$

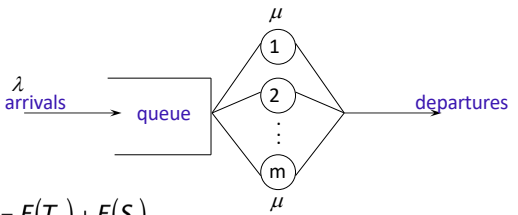
Prof. Vittoria de Nitto Personè

32

32

Analytical models
basic laws

Multi Server Queue



$$E(T_s) = E(T_q) + E(S_i)$$

Little's law

$$E(N_s) = \lambda E(T_s)$$

$$E(N_q) = \lambda E(T_q)$$

$$E(T_s) = \frac{E(N_s)}{\lambda}$$

$$E(T_q) = \frac{E(N_q)}{\lambda}$$

Prof. Vittoria de Nitto Personè

33