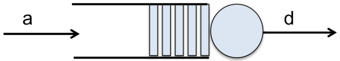
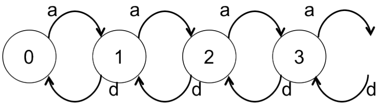
Queueing systems

Renato Lo Cigno

Simulation and Performance Evaluation 2018-19

- A Birth-Death process is a good model (and solver) of a queue





- Indeed queues can be used to model a variety of problems
  - CPUs, Stacks, Communication Links, ...
  - Post Offices, Banks, Offices in general, ...
  - Production plants, Logistics, ...
- Underlying a queuing system we always find a Markov Chain (DT, or CT, or Semi-Markov)

Kendall Notation

A queue is normally indicated with the following notation

$$A/S/m/B/K/SD$$

called the *Kendall notation* where

- A: defines the type of arrival
- S: defines the type of service
- m: defines the number of servers
- B: defines the maximum number of jobs/customers in the systems (including those in service) (omitted if ∞)
- K: defines the total population size (omitted if ∞)
- SD: defines the serving discipline (omitted if FCFS)

Arrival and Service processes (A/S)

- M: Markovian arrival/services, it means that interarrival times (service times) are exponentially distributed
- G: (General) arrival/services are arbitrarily distributed
- D: Deterministic
- E_k: arrival/services are Erlang with k stages
- H_k: arrival/services are Hyperexponential with k stages

Serving Disciplines

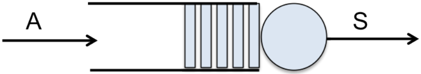
- FCFS (FIFO): First Come First Served
- LCFS (LIFO): Last Come First Served (stacks)
- PS: Processor Sharing
- R or SIRO: Service in Random Order
- PNPN: Priority Service (customers belong to classes) includes preemptive and non-preemptive systems (e.g., interrupts in OS and CPUs are –normally– preemptive)

- **M/M/1**
  Exponential interarrival times, exponential service times, 1 server, ∞ buffering positions, ∞ population, FCFS
- **M/G/2/PS**
  Exponential interarrival times, general service times, 2 servers, ∞ buffering positions, ∞ population, Processor Sharing
- **M/M/4/40/400/LIFO**
  Exponential interarrival times, exponential service times, 4 servers, 40 buffering positions, 400 potential customers/jobs, Last In First Out

- They model a wide range of systems
- Queues can be grouped in networks of queues and the solution remains an Markov Chain
- There are many "already solved" queues that we can use for quick-n-dirty evaluation
- There is a large class of networks of queues that allow a simple "product form solution"

- Number of customers in the queue
  - Easy as we associate the number of customers to the state of the MC so given the steady state distribution π of the MC representing the queuing system
    $\mathbf{P}[\text{No. of customers} = k] = \pi_k$
- Waiting times
- Average values (steady state analysis)
- Variance
- Distribution in steady state
- Transients (rarely)

Given a queue with general arrivals and services

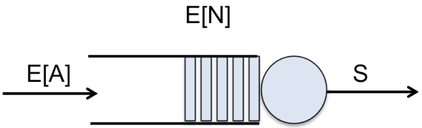


the average number of customers $E[N]$ is easily computed from the steady state π

$$E[N] = \sum_{k=0}^{\infty} k\pi_k$$

What if we want to know what is the average waiting (or response) time of the system $E[R]$?

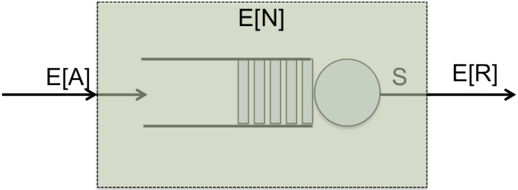
Given a queue without losses (either there are infinite position or B≥K)



with average arrival rate $E[A]$ and average number of customers $E[N]$, the average waiting time $E[R]$ is given by a very simple formula known as Little's formula

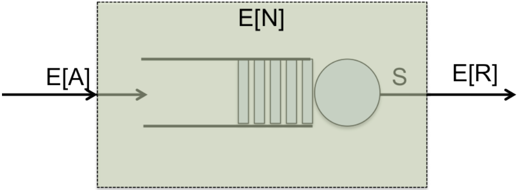
$$E[R] = \frac{E[N]}{E[A]}$$

Little's formula can be demonstrated based on conservation laws: whatever gets into a "black box" must come out



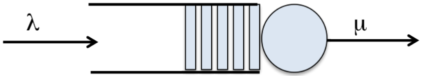
- The result is independent from: No. of servers, arrival distribution, and service distribution

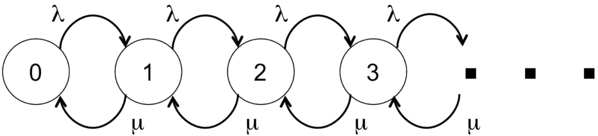
Interpretation of Little's result



- States that the expected waiting time is directly proportional to the number of customers in the system and inversely proportional to the average arrival rate
- The result is independent from the service distribution, but it requires that $E[A] < E[S]$
- *The system must be without losses*

- All CTMCs underlying continuous time queues with Markovian arrival and services are Birth-Death processes
- In general the steady-state solution is not difficult to compute
- We call λ the average arrival rate
- We call μ the service rate of a single server
- We call $\rho = \frac{\lambda}{\mu}$ the *load* of the queue
- The infinitesimal generator Q is diagonal or banded


λ
μ


λ
λ
λ
λ
0
1
2
3
μ
μ
μ
μ

- Must be $\rho < 1$ for stability
- The general balance requires $\lambda\pi_i = \mu\pi_{i+1}$ or $\pi_{i+1} = \rho\pi_i$
- By direct substitution we have

$$\pi_i = \rho^i \pi_0; \quad i > 0; \quad \text{and} \quad \sum_{i=0}^{\infty} \pi_i = 1$$

$$\pi_0 = \left[\sum_{i=0}^{\infty} \rho^i\right]^{-1} = (1-\rho)$$

$$\pi_i = (1-\rho)\rho^i$$

- The average number of customer is

$$E[N] = \sum_{i=0}^{\infty} i\pi_i = (1-\rho)\sum_{i=0}^{\infty} i\rho^i = \frac{\rho}{1-\rho}$$

- The variance of the number of customer is

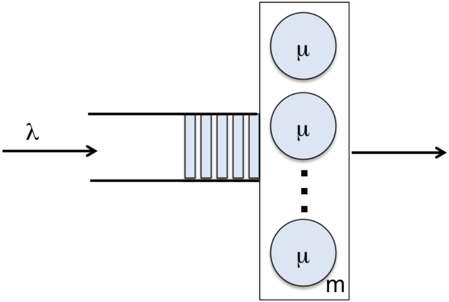
$$\mathrm{Var}[N] = \sum_{i=0}^{\infty} i^2\pi_i - (E[N])^2 = \frac{\rho}{(1-\rho)^2}$$

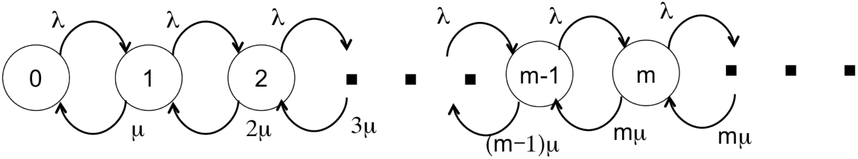
- And applying Little's rule we obtain the average waiting time

$$E[R] = \frac{E[N]}{\lambda} = \frac{\rho}{\lambda(1-\rho)} = \frac{1/\mu}{1-\rho}$$

note that it is the average service time over the probability that the server is idle

- Homework: plot $E[N]$ and $E[R]$ as a function of ρ


λ
μ
μ
μ
m


λ
λ
λ
λ
λ
λ
0
1
2
m-1
m
μ
2μ
3μ
(m−1)μ
mμ
mμ

- Must be $\rho < m$ for stability
- The general balance equations a simple but a little cumbersome, as they have to include the varying service rate for $i < m$, so we only give the final results

$$\pi_0 = \left[\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!} \frac{1}{1-\rho}\right]^{-1}$$

$$\pi_i = \pi_0 \rho^i \frac{1}{m!}; \quad i \leq m$$

$$\pi_i = \pi_0 \rho^i \frac{1}{m! m^{m-i}}; \quad i \geq m$$

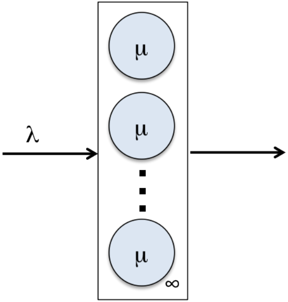
- The average number of customer is

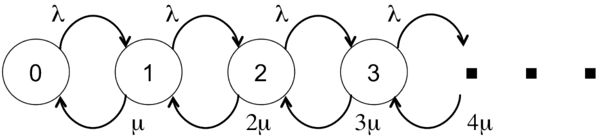
$$E[N] = \sum_{i=0}^{\infty} i\pi_i = m\rho + \rho \frac{(m\rho)^m}{m!} \frac{\pi_0}{(1-\rho)^2}$$

- And applying Little's rule we obtain the average waiting time

$$E[R] = \frac{E[N]}{\lambda} = m\frac{1}{\mu} + \frac{1}{\mu} \frac{(m\rho)^m}{m!} \frac{\pi_0}{(1-\rho)^2}$$

M/M/∞





M/M/∞

- The queue is always stable for $\rho < \infty$
- The general balance requires $\lambda\pi_i = (i+1)\mu\pi_{i+1}$ or $\pi_{i+1} = \frac{\rho}{i+1}\pi_i$
- By direct substitution we have

$$\pi_i = \frac{\rho^i}{i!}\pi_0; \quad i > 0; \quad \text{and} \quad \sum_{i=0}^{\infty}\pi_i = 1$$

$$\pi_0 = \left[\sum_{i=0}^{\infty}\frac{\rho^i}{i!}\right]^{-1} = e^{-\rho}$$

$$\pi_i = \frac{\rho^i}{i!}e^{-\rho}$$

- The average number of customer is

$$E[N] = \sum_{i=0}^{\infty} i\pi_i = \rho$$

- The variance of the number of customer is

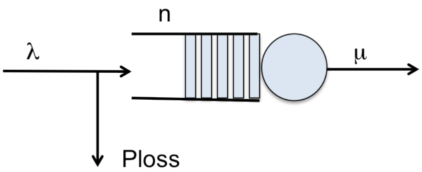
$$\begin{aligned} \mathrm{Var}[N] &= \sum_{i=0}^{\infty} i^2\pi_i - (E[N])^2 = e^{-\rho} \sum_{i=0}^{\infty} i^2 \frac{\rho^i}{i!} - \rho^2 \\ &= e^{-\rho} e^{\rho}(\rho + \rho^2) - \rho^2 = \rho \end{aligned}$$

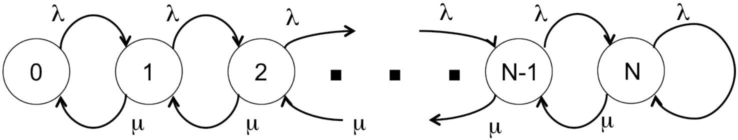
- As there is no queuing (infinite servers) we don't even need Little's rule to obtain the average response time

$$E[R] = \frac{1}{\mu}$$

Compare the performance in terms of average number of customers and average response time of the following three queuing systems

- M/M/1 with service rate $m\mu$ and arrival rate $m\lambda$
- M/M/m with service rate μ and arrival rate $m\lambda$
- m parallel M/M/1 queues with service rate μ and arrival rate λ


n
λ
μ
Ploss


λ
λ
λ
λ
λ
λ
0
1
2
N-1
N
μ
μ
μ
μ
μ

- A finite queue does not have stability problems, so $0 < \rho < \infty$
- The general balance requires $\lambda\pi_i = \mu\pi_{i+1}$ or $\pi_{i+1} = \rho\pi_i$
- When new arrivals happen in state n the customers are lost
- By direct substitution we have

$$\pi_i = \rho^i \pi_0; \quad 0 < i < n; \quad \text{and} \quad \sum_{i=0}^{n} \pi_i = 1$$

$$\pi_0 = \left[\sum_{i=0}^{n} \rho^i\right]^{-1} = \begin{cases} \dfrac{1-\rho}{1-\rho^{n+1}}; & \rho \neq 1 \\ \dfrac{1}{n+1}; & \rho = 1 \end{cases}$$

- The loss probability is given by the probability that a customer arrives in state N conditioned on the probability that a customer has arrived, so it is simply

$$P_{\text{loss}} = \pi_n = \frac{1-\rho}{1-\rho^{n+1}}\rho^n$$

- P_{loss} is always smaller than the probability that the queue length in and M/M/1 queue is larger or equal to n
- The reason is that a queuing customers creates a dependence or correlation in time equal to its service time that is paid by all customers that arrive later, while refusing a customer is terms of service time is equal to 0
- Homework: prove it or show it graphically for different $\rho < 1$

- Average number of customers

$$
\begin{aligned}
E[N] &= \sum_{i=1}^{n} i\pi_i \\
&= \sum_{i=1}^{\infty} i\pi_i - \sum_{i=n+1}^{\infty} i\pi_i \\
&= \frac{\rho}{1-\rho} - \frac{n+1}{1-\rho^{n+1}}\rho^{n+1}
\end{aligned}
$$

Throughput of the queue

- For an infinite queuing system the notion of throughput is meaningless: everything that comes in must exit
- If there are losses instead we can be interested to know what it the number (fraction) of customers serviced
- Intuitively this is the total minus the lost ones so $Th = \lambda(1 - \pi_n)$
- Also intuitively it should be one minus the time the server is inactive, hence $Th = \mu(1 - \pi_0)$
- Interestingly this also means that $\dfrac{(1 - \pi_0)}{(1 - \pi_n)} = \rho$

In general the throughput can be computed as the arrival rate in any state that does not lead to a loss

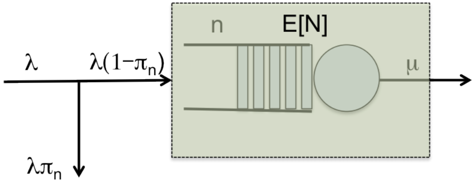
- For a generic (finite) queue with one server

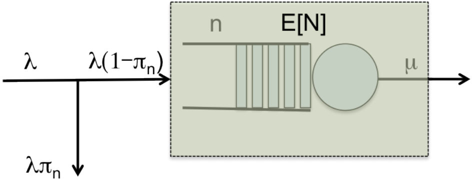
$$Th = \sum_{i=0}^{n-1} \lambda_i \pi_i$$

note that π is not necessarily simple to compute

Response time in finite queues

- Little's result cannot be applied directly because there are losses
- However we know what are the losses and hence the net flow entering the queue after the lost customers are discarded





- Litte's result can be applied to this subsystem

$$E[R] = \frac{E[N]}{Th} = \frac{1}{\lambda(1-\pi_n)}\left[\frac{\rho}{1-\rho} - \frac{n+1}{1-\rho^{n+1}}\rho^{n+1}\right]$$

- If we can compute the loss probability and hence the throughput, then Little's formula can be applied (any system, not only the M/M/1/n)

- We can imagine all sort of single station queuing systems
  - With non Markov arrivals/services
  - With batch arrivals
  - With servers that sometimes stop serving
  - ...
- Many have closed form or approximate solutions
- Some are important
- Finding the solution is often complex ...

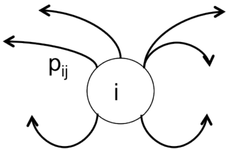
Given an M/M/1 queue

- All results are stochastically independent from the serving policy
- LIFO, Random, PS, ...
- As long as it is *work conserving*

- The result is intuitive as all customers are identical and their service bears no memory, so even a policy that tries to favor someone is impossible

- Can we find results for non-markovian services?
- Indeed yes, using a very interesting technique: Using a discrete MC obtained sampling the system at times where all the memory is embedded in the state
- But what are these times?
- The problem lies in the fact that the residual service time is not independent from the service already received
- But different customers are independent one another ...

- . . . if we sample the system when a customer departs, then we obtain a DTMC . . .
- . . . and we are left (only!) with the problem of computing the transition probabilities p_{ij}



Let

- $X = X_n;\ \ n = 0, 1, 2, 3, \cdots$ be the (DT) stochastic process that describes the number of customers in the queue at the departure of the *n*-th customer, and
- $Y = Y_n;\ \ n = 0, 1, 2, 3, \cdots$ be the (DT) stochastic process that describes the number of customers that arrive during the service of the *n*-th customer

then we have

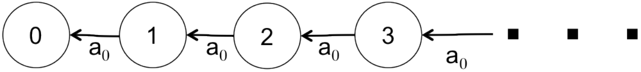
$$X_{i+1} = \begin{cases} X_i - 1 + Y_{i+1}, & \text{if } X_i > 0 \\ Y_{i+1}, & \text{if } X_i = 0 \end{cases}$$

$$X_{i+1} = \begin{cases} X_i - 1 + Y_{i+1}, & \text{if } X_i > 0 \\ Y_{i+1}, & \text{if } X_i = 0 \end{cases}$$

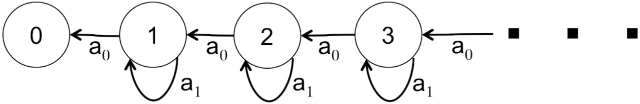
- The case for $X_i > 0$ is straightforward: when customer $i+1$ leaves the system he leaves behind the customers that were in the queue when his service started, minus himself, plus the customers arrived during its service
- The case for $X_i = 0$ goes as follows: when customer $i+1$ leaves the system he has first arrived, so the queue that was empty now has a customer, but then he leaves, so he leaves the customers arrived during its service

M/G/1

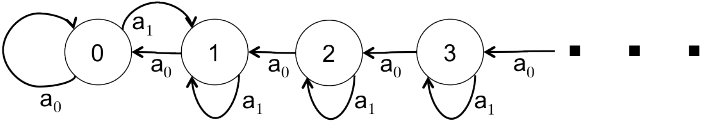
- What are the possible events when a customer departs?
- What states can be reached with these events?
- What are the probabilities of these events?



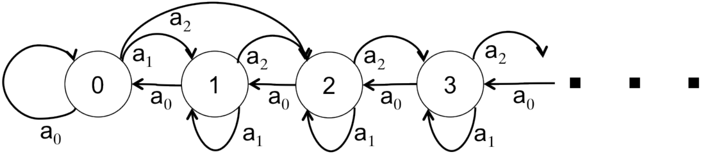
- First event: no customer arrives during a service
- Clearly this means a transition $i \to i-1;\ i > 0$
- Let's call the probability of this event a_0



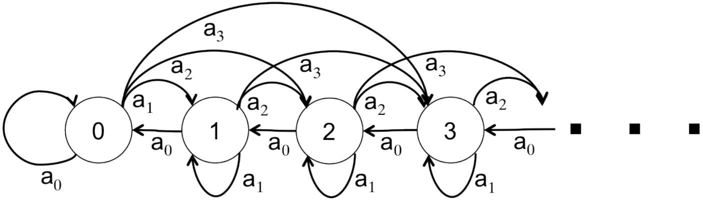
- Second event: one customer arrives during a service
- Clearly this means a transition $i \to i;\ i > 0$, as the additional customer compensate the one leaving on service completion
- Let's call this probability a_1



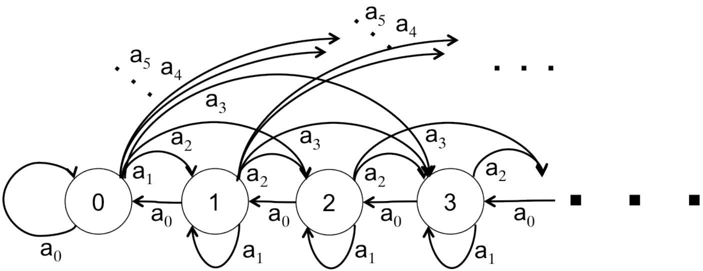
- But what about transitions from state 0?
- We have no customer in state 0, so transition $0 \to 0$ means that a customer has arrived, and then no other has arrived until he left
- Then this transition has probability a_0
- In general transitions $0 \to j$ happen with probability a_j that j new customers arrive during the service of the customer that arrived and has been served



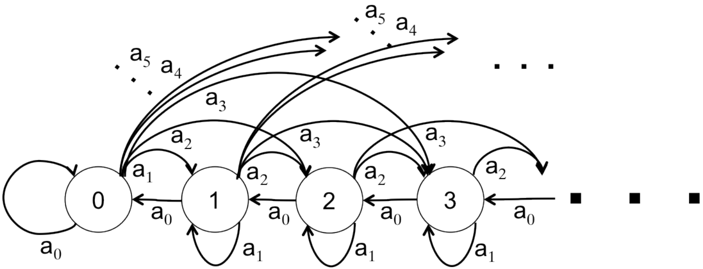
- Third event: two customers arrive during a service
- The transition is $i \rightarrow i+1;\ i \geq 0$ as one additional customer compensate the one leaving on service completion and the second one increase the no. of customers in the queue by 1
- Or transition $0 \rightarrow 2$
- We call this probability a_2



- Fourth event: three customers arrive during a service
- This means a transition $i \to i+2;\ i \geq 0$ or $0 \to 3$
- We call this probability a_3



- We can recursively continue the reasoning to obtain all the infinite a_j transition probabilities from a given state to the others



- Notice that the transition probabilities from any state is identical to any other state
- With the exception of state 0 where it is not possible to have a transition to state "-1" and actually the probability a_0 that no customer arrives during a service is added to the self-transition

The one step transition probability matrix is thus

$$P = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & \cdots \\ a_0 & a_1 & a_2 & a_3 & a_4 & \cdots \\ 0 & a_0 & a_1 & a_2 & a_3 & \cdots \\ 0 & 0 & a_0 & a_1 & a_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

- The structure is known as *upper Hassenberg* (all values below the first lower sub-diagonal are zero) and the system can be solved with known techniques (e.g., QR-decomposition) by reducing it to a triangular matrix
- Still we have to formalize the a_j

Since arrivals follow a Poisson process, then in general, if we call B the RV describing the services we can write

$$\mathbf{P}[Y_{n+1} = j | B = t] = e^{-\lambda t} \frac{(\lambda t)^j}{j!}$$

Applying the theorem of total probability

$$a_j = \int_0^\infty \mathbf{P}[Y_{n+1} = j | B = t] \, f_B(t) \, dt = \int_0^\infty e^{-\lambda t} \frac{(\lambda t)^j}{j!} \, f_B(t) \, dt$$

- a_j can be computed once the distribution $f_B(t)$ of the services is given

M/G/1 – computing $E[N]$

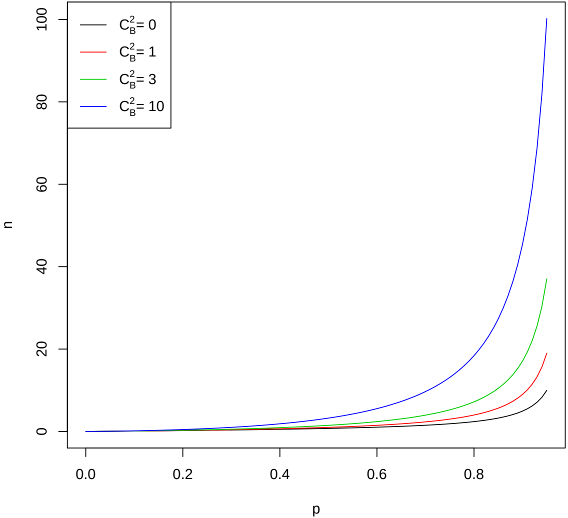
- Completing the analysis of the M/G/1 queue, once we found the one step transition probability matrix is still difficult and requires math manipulations in the z (discrete frequency transform domain) and LS (Laplace Stilties) domains, which we do not know ...
- ... after these passages, however we can come to the very general and powerful result giving the average expected number of customers $E[N]$ for any queuing system

$$E[N] = \rho + \frac{\rho^2}{2(1-\rho)}(1 + C_B^2)$$

where $C_B = \frac{\sigma_B}{\mu_B}$ is the coefficient of variation of the service time distribution

$$E[N] = \rho + \frac{\rho^2}{2(1-\rho)}(1 + C_B^2) = \frac{\rho}{1-\rho}(1 + \rho\frac{C_B^2 - 1}{2})$$

- Given a load ρ, $E[N]$ grows linearly with C_B^2
- $E[N]$ depends only on the first two moments of the services distribution
- If $C_B^2 \to \infty$ then also $E[N]$ goes to infinity: the queuing system "seems" stable, but it response time becomes infinite



C_B^2= 0
C_B^2= 1
C_B^2= 3
C_B^2= 10
n
0
20
40
60
80
100
0.0
0.2
0.4
0.6
0.8
p

- A very interesting service discipline is the Processor Sharing (PS) that approximate a round robin (RR) discipline as the service time in the RR discipline approaches 0 and the RR overhead is negligible
- Jobs enter in service as soon as they arrive, but if there are j customers each job receives only $\frac{1}{j}$ of the processing power
- Intuitively this serving discipline favors short jobs that will stay in the system for a short time, while long jobs will stay in the system for a very long time, as they are continuously "disturbed" (i.e., the processing power dedicated to them is reduced) by short jobs arriving in the system
- The formal analysis is not trivial

- The analysis show that "average" performance of the M/G/1/PS queue are the same of the M/M/1/FIFO, a very notable result that also tell and support the intuition that if the service is "shared" then there is no blocking phenomenon as, instead, happens in the M/G/1/FIFO for $C_B \to \infty$
- Distributions however are not, indeed distributions are even more biased and "stretched" if $C_B > 1$ as heavy jobs remain in the system for very long time
- PS discipline create "long range dependence," i.e., a single large job introduces a long memory in the system, even if arrival and services are memory-less.

- $E[S] = 1/\mu$: average (total) service time per job; $\rho = \lambda/\mu$
- $\pi_0 = 1 - \rho$
- $\pi_i = (1 - \rho)\rho^i$
- $E[N] = \dfrac{\rho}{1 - \rho}$
- $\text{Var}[N] = \dfrac{\rho}{(1 - \rho)^2}$
- $E[R] = \dfrac{1}{\mu(1 - \rho)}$ the queuing delay is not defined