Queueing Theory

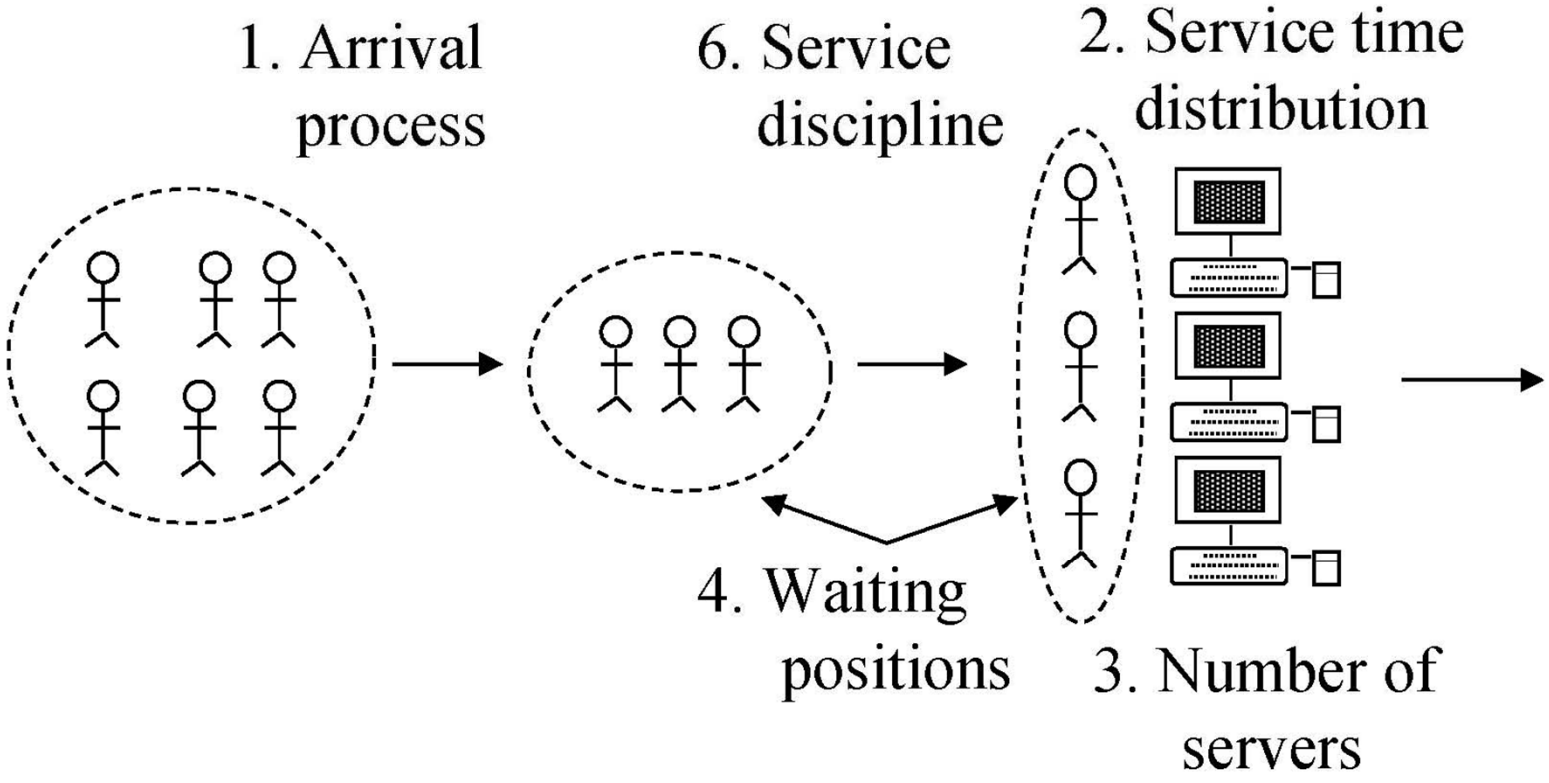
Overview

- Queuing Notation
- Rules for All Queues
- Little's Law

What you will learn

- What are various types of queues.
- What is meant by an *M/M/m/B/K* queue?
- How to obtain response time, queue lengths, and server utilizations?
- How to represent a system using a network of several queues?
- How to analyze simple queuing networks?
- How to obtain bounds on the system performance using queuing models?
- How to obtain variance and other statistics on system performance?
- How to subdivide a large queuing network model and solve it?

Basic Components of a Queue



Kendall Notation *A/S/m/B/K/SD*

- *A*: Arrival process
- *S*: Service time distribution
- *m*: Number of servers
- *B*: Number of buffers (system capacity)
- *K*: Population size, and
- *SD*: Service discipline

Arrival Process

- Arrival times: $t_1, t_2, \ldots, t_j$
- Interarrival times: $\tau_j = t_j - t_{j-1}$
- τ_j form a sequence of Independent and Identically Distributed (IID) random variables
- Exponential + IID $\Rightarrow$ Poisson
- Notation:
  - M = Memoryless = Poisson
  - E = Erlang
  - H = Hyper-exponential
  - G = General $\Rightarrow$ Results valid for all distributions

Service time distribution

- Time each student spends at the terminal.
- Service times are IID.
- Distribution: M, E, H, or G
- Device = Service center = Queue
- Buffer = Waiting positions

Services disciplines

- First-Come-First-Served (FCFS)
- Last-Come-First-Served (LCFS)
- Last-Come-First-Served with Preempt and Resume (LCFS-PR)
- Round-Robin (RR) with a fixed quantum.
- Small Quantum ⇒ Processor Sharing (PS)
- Infinite Server: (IS) = fixed delay
- Shortest Processing Time first (SPT)
- Shortest Remaining Processing Time first (SRPT)
- Shortest Expected Processing Time first (SEPT)
- Shortest Expected Remaining Processing Time first (SERPT).
- Biggest-In-First-Served (BIFS)
- Loudest-Voice-First-Served (LVFS)

Common Distributions

- *M*: Exponential
- E_k: Erlang with parameter *k*
- H_k: Hyper-exponential with parameter *k*
- *D*: Deterministic ⇒ constant
- *G*: General ⇒ All
- Memory less:
  - Expected time to the next arrival is always $1/\lambda$ regardless of the time since the last arrival
  - Remembering the past history does not help.

Example M/M/3/20/1500/FCFS

- Time between successive arrivals is exponentially distributed.
- Service times are exponentially distributed.
- Three servers
- *20* Buffers = *3* service + *17* waiting
- After *20*, all arriving jobs are lost
- Total of *1500* jobs that can be serviced.
- Service discipline is first-come-first-served.
- Defaults:
    - Infinite buffer capacity
    - Infinite population size
    - FCFS service discipline.
- *G/G/1 = G/G/1/ ∞/ ∞/ FCFS*

Quiz (1)

- True or false

❑The numbers of servers in a M/M/1/3 queue is 3

❑G/G/1/30/300/LCFS queue is like stack

❑M/D/3/30 queue has 30 buffers

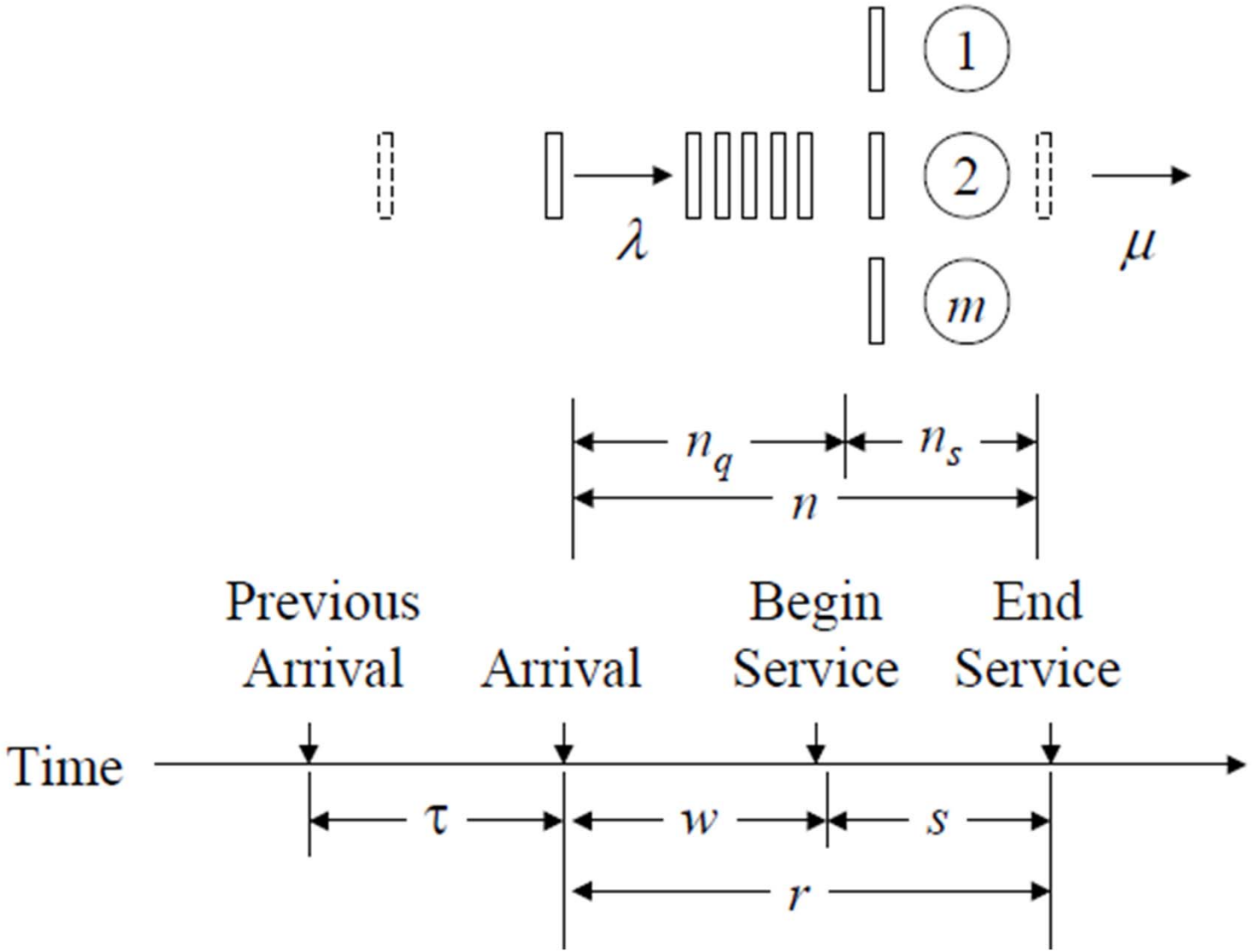
❑G/G/1 queue has ∞ population

❑D/D/1 queue has FCFS discipline

Quiz (2)

- Exponential distribution is denoted as ---------
- ----------distribution represents a set of parallel exponential servers
- Erlang distribution E_k with k=1 is same as ---------- distribution

Key Variables



Key Variables (cont)

- τ = Inter-arrival time = time between two successive arrivals.
- λ = Mean arrival rate = $1/E[\tau]$
  May be a function of the state of the system,
  e.g., number of jobs already in the system.
- s = Service time per job.
- μ = Mean service rate per server = $1/E[s]$
- Total service rate for m servers is $m\mu$
- n = Number of jobs in the system.
  This is also called **queue length**.
- Note: Queue length includes jobs currently receiving service as well as those waiting in the queue.

Key Variables (cont)

- n_q = Number of jobs waiting
- n_s = Number of jobs receiving service
- r = Response time or the time in the system
  = time waiting + time receiving service
- w = Waiting time
  = Time between arrival and beginning of service

Rules for All Queues

Rules: The following apply to *G/G/m* queues

1. Stability Condition:

$$\lambda < m\mu$$

Finite-population and the finite-buffer systems are always stable.

2. Number in System versus Number in Queue:

$n = n_q + n_s$

Notice that n, n_q, and n_s are random variables.

$E[n]=E[n_q]+E[n_s]$

If the service rate is independent of the number in the queue,

$Cov(n_q,n_s) = 0$

$$\mathrm{Var}[n] = \mathrm{Var}[n_q] + \mathrm{Var}[n_s]$$

Rules for All Queues (cont)

3. Number versus Time:

   If jobs are not lost due to insufficient buffers,

   Mean number of jobs in the system

   = Arrival rate × Mean response time

4. Similarly,

   Mean number of jobs in the queue

   = Arrival rate × Mean waiting time

This is known as **Little's law**.

5. Time in System versus Time in Queue

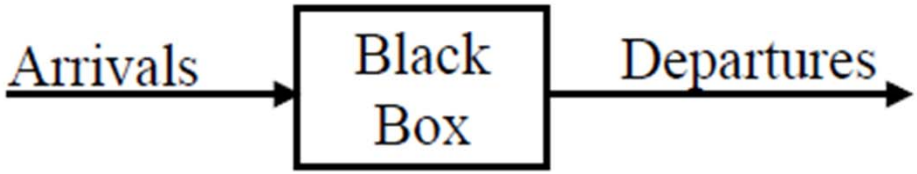
$$r = w + s$$

r, w, and s are random variables.

$$E[r] = E[w] + E[s]$$

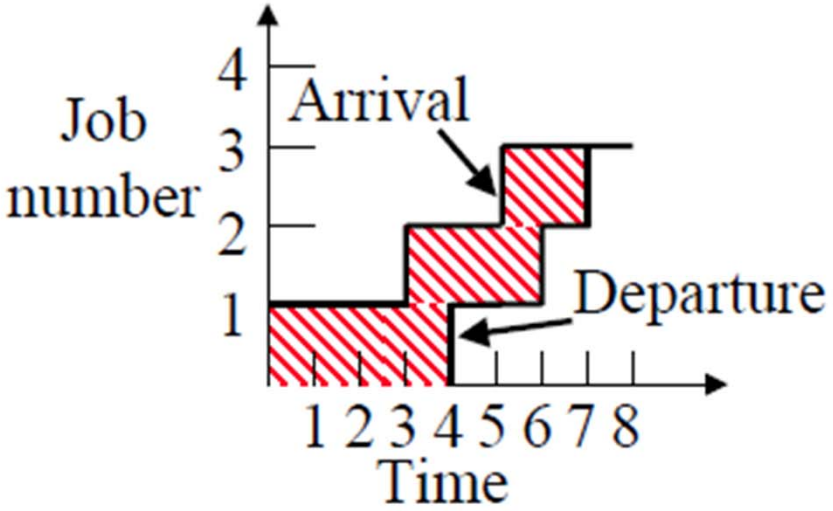
Little's Law

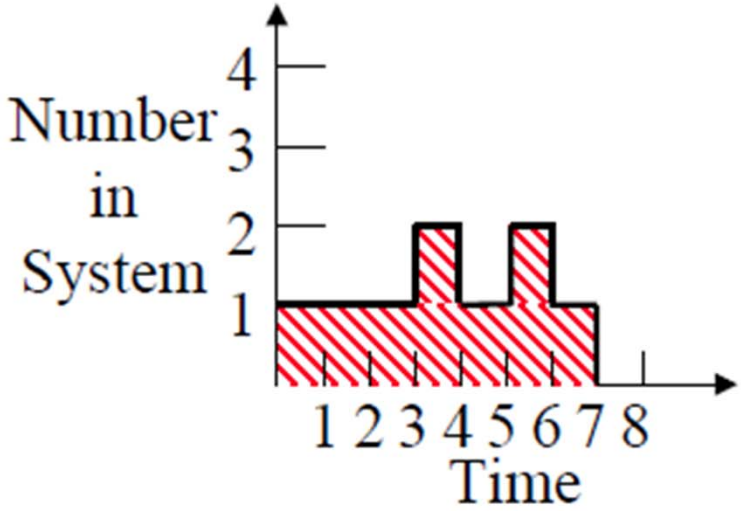
- Mean number in the system
  = Arrival rate × Mean response time
- This relationship applies to all systems or parts of systems in which the number of jobs entering the system is equal to those completing service.
- Named after Little (1961)
- Based on a black-box view of the system:



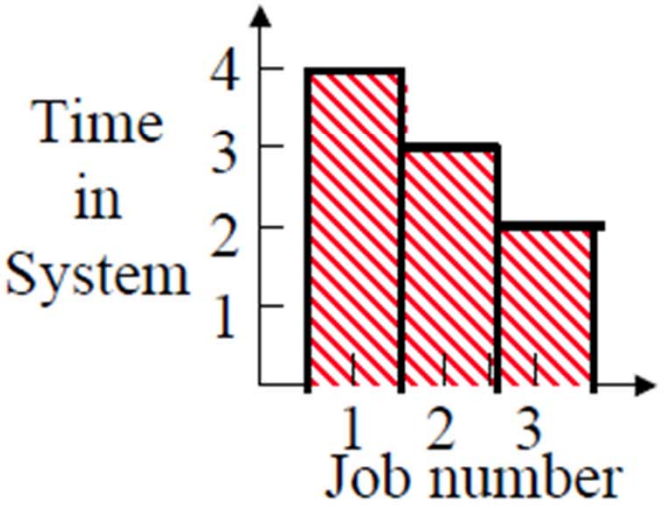
- In systems in which some jobs are lost due to finite buffers, the law can be applied to the part of the system consisting of the waiting and serving positions

Proof of Little's Law

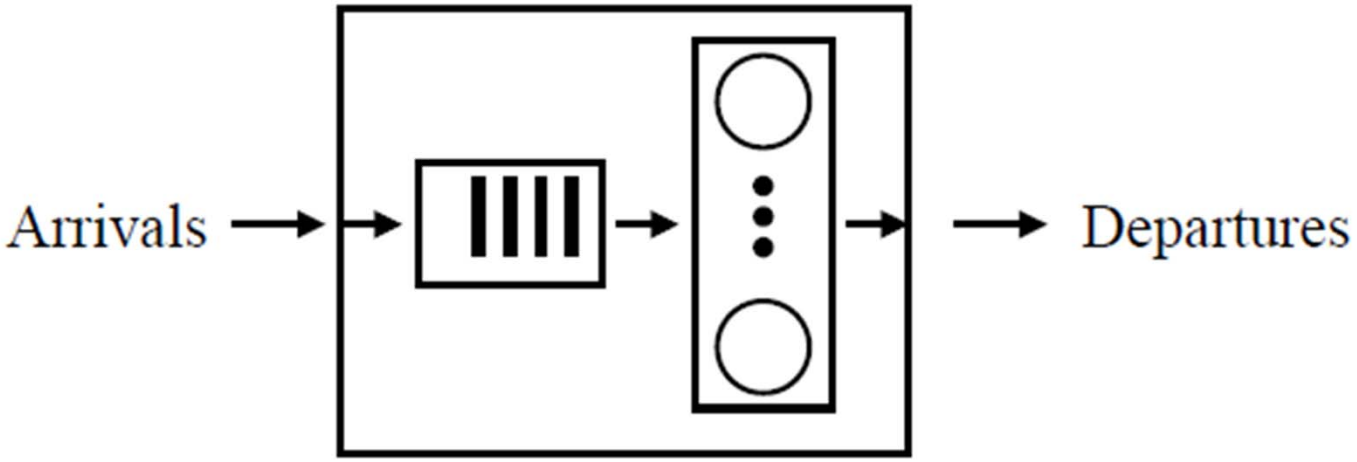




- If T is large, arrivals = departures = N
- Arrival rate = Total arrivals/Total time= N/T
- Hatched areas = total time spent inside the system by all jobs = J
- Mean time in the system= J/N
- Mean Number in the system
  $= J/T = \frac{N}{T} \times \frac{J}{N}$
  = Arrival rate× Mean time in the system



Application of Little's Law



- Applying to just the waiting facility of a service center
- Mean number in the queue = Arrival rate × Mean waiting time
- Similarly, for those currently receiving the service, we have:
- Mean number in service = Arrival rate × Mean service time

example

- A monitor on a disk server showed that the average time to satisfy an I/O request was 100 milliseconds. The I/O rate was about 100 requests per second. What was the mean number of requests at the disk server?
- Using Little's law:
- Mean number in the disk server = Arrival rate × Response time
- = 100 (requests/second) ×(0.1 seconds)
- = 10 requests

Quiz

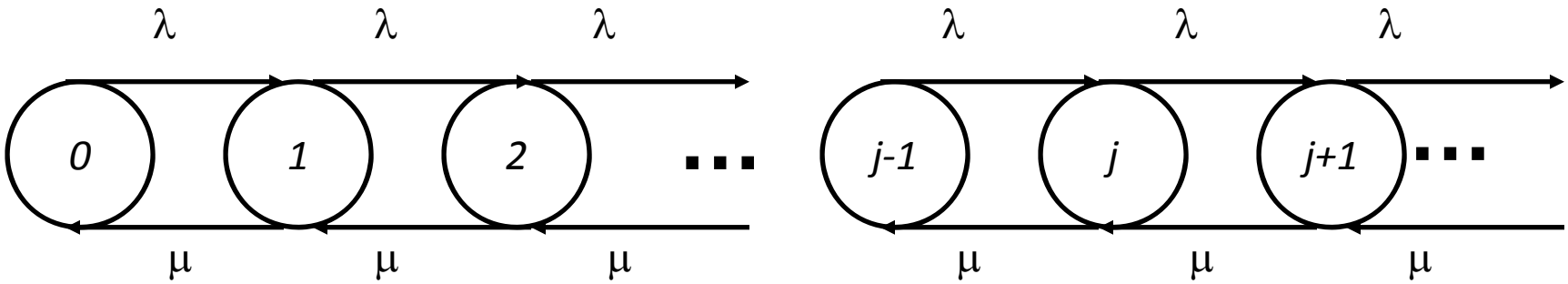
- If a queue has 2 persons waiting for service, the number is system is ----------
- If the arrival rate is 2 jobs/second, the mean inter-arrival tie is ------- second
- In a 3 server queue, the jobs arrive at the rate of 1 jobs/second, the service time should be less than ----- second/job for the queue to be stable
- During a 1 minutes observation, a server received 120 requests. The mean response time was 1 second. The mean number of queries in the server is --------

Home work

- During a one-hour observation interval, the name server of a distributed system received *10,800* requests. The mean response time of these requests was observed to be one-third of a second. What is the mean number of queries in the server? What assumptions have you made about the system? Would the mean number of queries be different if the service time was not exponentially distributed?

M/M/1 Queue

- *M/M/1* queue is the most commonly used type of queues
- Used to model single processor systems or to model individual devices in a computer system
- Assumes that the interarrival times and the service times are exponentially distributed and there is only one server.
- No buffer or population size limitations and the service discipline is FCFS
- Need to know only the mean arrival rate λ and the mean service rate μ.
- State = number of jobs in the system



Results for M/M/1 Queue

- Birth-death processes with

$$\lambda_n = \lambda \quad n = 0, 1, 2, \ldots, \infty$$

$$\mu_n = \mu \quad n = 1, 2, \ldots, \infty$$

- Probability of *n* jobs in the system:

$$p_n = \left(\frac{\lambda}{\mu}\right)^n p_0 \quad n = 1, 2, \ldots, \infty$$

Results for M/M/1 Queue (Cont)

- The quantity λ/μ is called traffic intensity and is usually denoted by the symbol ρ. Thus:

$$p_n = \rho^n p_0$$

$$p_0 = \frac{1}{1 + \rho + \rho^2 + \cdots + \rho^\infty} = 1 - \rho$$

$$p_n = (1 - \rho)\rho^n \quad n = 0, 1, 2, \ldots, \infty$$

- *n* is geometrically distributed.
- Utilization of the server
  = Probability of having one or more jobs in the system:

$$U = 1 - p_0 = \rho$$

Results for M/M/1 Queue (Cont)

- Mean number of jobs in the system:

$$E[n] = \sum_{n=1}^{\infty} np_n = \sum_{n=1}^{\infty} n(1-\rho)\rho^n = \frac{\rho}{1-\rho}$$

- Variance of the number of jobs in the system:

$$\begin{aligned} \mathrm{Var}[n] &= E[n^2] - (E[n])^2 \\ &= \left(\sum_{n=1}^{\infty} n^2(1-\rho)\rho^n \right) - (E[n])^2 = \frac{\rho}{(1-\rho)^2} \end{aligned}$$

Results for M/M/1 Queue (Cont)

- Probability of *n* or more jobs in the system:

$$P(\geq n \text{ jobs in system}) = \sum_{j=n}^{\infty} p_j = \sum_{j=n}^{\infty} (1-\rho)\rho^j = \rho^n$$

- Mean response time (using the Little's law):

Mean number in the system = Arrival rate $\times$ Mean response time

That is:

$$E[n] = \lambda E[r]$$

$$E[r] = \frac{E[n]}{\lambda} = \left(\frac{\rho}{1-\rho}\right)\frac{1}{\lambda} = \frac{1/\mu}{1-\rho}$$

Results for M/M/1 Queue (Cont)

- Mean number of jobs in the queue:

$$E[n_q] = \sum_{n=1}^{\infty}(n-1)p_n = \sum_{n=1}^{\infty}(n-1)(1-\rho)\rho^n = \frac{\rho^2}{1-\rho}$$

- The probability of blocking or lost
- $$B(x) = \sum_{i=k}^{\infty} P_i = \sum_{i=k}^{\infty}(1-\rho)P_i = \rho^k(1-\rho)\sum_{i=k}^{\infty} P_i = \rho^k$$

  $$B(x) = \rho^k$$
- Idle $\Rightarrow$ there are no jobs in the system
- Busy period = The time interval between two successive idle intervals

Example

- On a network gateway, measurements show that the packets arrive at a mean rate of 125 packets per second (pps) and the gateway takes about two milliseconds to forward them. Using an M/M/1 model, analyze the gateway. What is the probability of buffer overflow if the gateway had only 13 buffers? How many buffers do we need to keep packet loss below one packet per million?
  - Arrival rate λ = 125 pps
  - Service rate μ = 1/.002 = 500 pps
  - Gateway Utilization $\rho = \lambda/\mu = 0.25$
  - Probability of *n* packets in the gateway $= (1-\rho)\rho^n = 0.75(0.25)^n$

Example (Cont)

- Mean Number of packets in the gateway
  = $\rho/(1-\rho) = 0.25/0.75 = 0.33$
- Mean time spent in the gateway
  = $(1/\mu)/(1-\rho) = (1/500)/(1-0.25) = 2.66$ milliseconds
- Probability of buffer overflow

  P(more than 14 packets in the gateway)

  = $\rho^{14} = 0.25^{14} = 3.73 * 10^{-9}$

  ≈ 4 packets per billion packets
- To limit the probability of loss to less than 10^{-6}:

$$\rho^n \leq 10^{-6} \qquad n > \log(10^{-6})/\log(0.25) = 9.96$$

We need about ten buffers.

Example (Cont)

- The last two results about buffer overflow are approximate. Strictly speaking, the gateway should actually be modeled as a finite buffer *M/M/1/B* queue.

- However, since the utilization is low and the number of buffers is far above the mean queue length, the results obtained are a close approximation.

M/M/1

- During a one-hour observation interval, the name server of a distributed system received *10,800* requests. The mean response time of these requests was observed to be one-third of a second. What is the mean number of queries in the server? What assumptions have you made about the system? Would the mean number of queries be different if the service time was not exponentially distributed?

M/M/1 Queue

- *M/M/1* queue is the most commonly used type of queues
- Used to model single processor systems or to model individual devices in a computer system
- Assumes that the inter arrival times and the service times are exponentially distributed and there is only one server.
- No buffer or population size limitations and the service discipline is FCFS
- Need to know only the mean arrival rate λ and the mean service rate μ.
- State = number of jobs in the system