CPSC 531: System Modeling and Simulation

Carey Williamson
Department of Computer Science
University of Calgary

Fall 2017

Motivating Quote for Queueing Models

"Good things come to those who wait"

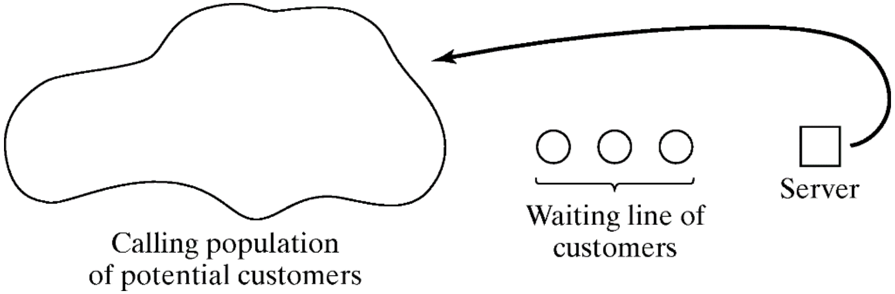
- poet/writer Violet Fane, 1892
- song lyrics by Nayobe, 1984
- motto for Heinz Ketchup, USA, 1980's
- slogan for Guinness stout, UK, 1990's

Outline: Queueing Models

1. Basic components of a queue
2. General rules
3. Markov models
4. Common queueing models

Queueing Systems

- A variety of systems can be modeled as a queue.
- A simple but typical queueing model:



- Queueing models provide the analyst with a powerful tool for designing and evaluating the performance of queueing systems
- Typical measures of system performance:
  - Server utilization, length of waiting lines, and delays of customers
  - For relatively simple systems, compute results mathematically
  - For realistic models of complex systems, simulation is usually required

Queueing Systems

- Key elements of queueing systems:
  - Customer: refers to anything that arrives at a facility and requires service (e.g., people, machines, trucks, emails)
  - Server: refers to any resource that provides the requested service (e.g., barber, repair person, car wash, file server)

Calling Population

- Calling population: the population of potential customers, which could be finite or infinite
  - Finite population model: arrival rate depends on the number of customers in the system, and their current states (e.g., if you have only one laptop, and it is currently at the repair shop, then the arrival rate of failed laptops from you becomes zero)
  - Infinite population model: arrival rate is not affected by the number of customers in the system (e.g., systems with large population of potential customers)

System Capacity

- System capacity: a limit on the maximum number of customers that may be in service or waiting in line
  - Limited capacity (e.g., an automatic car wash only has room for 10 cars to wait in line to enter the wash bay)
  - Unlimited capacity (e.g., concert ticket sales with no limit on the number of people allowed to wait to purchase tickets)

Queue Behavior and Queue Discipline

- Queue behavior: the actions of customers while in a queue waiting for service to begin, for example:
  - Balk: leave when they see that the line is too long
  - Renege: leave after being in the line when its moving too slowly
  - Jockey: move from one line to a shorter line

- Queue discipline: the logical ordering of customers in a queue that determines which customer is chosen for service when a server becomes available, for example:
  - First-in-first-out (FIFO)
  - Last-in-first-out (LIFO)
  - Service in random order (SIRO)
  - Shortest job first (SJF)
  - Service according to priority (PRI)

Queueing Notation

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

- ***A*** : Arrival process
- ***S*** : Service process
- ***m*** : Number of servers
- ***B*** : System capacity (finite buffer)
- ***K*** : Population size
- ***SD*** : Service discipline

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

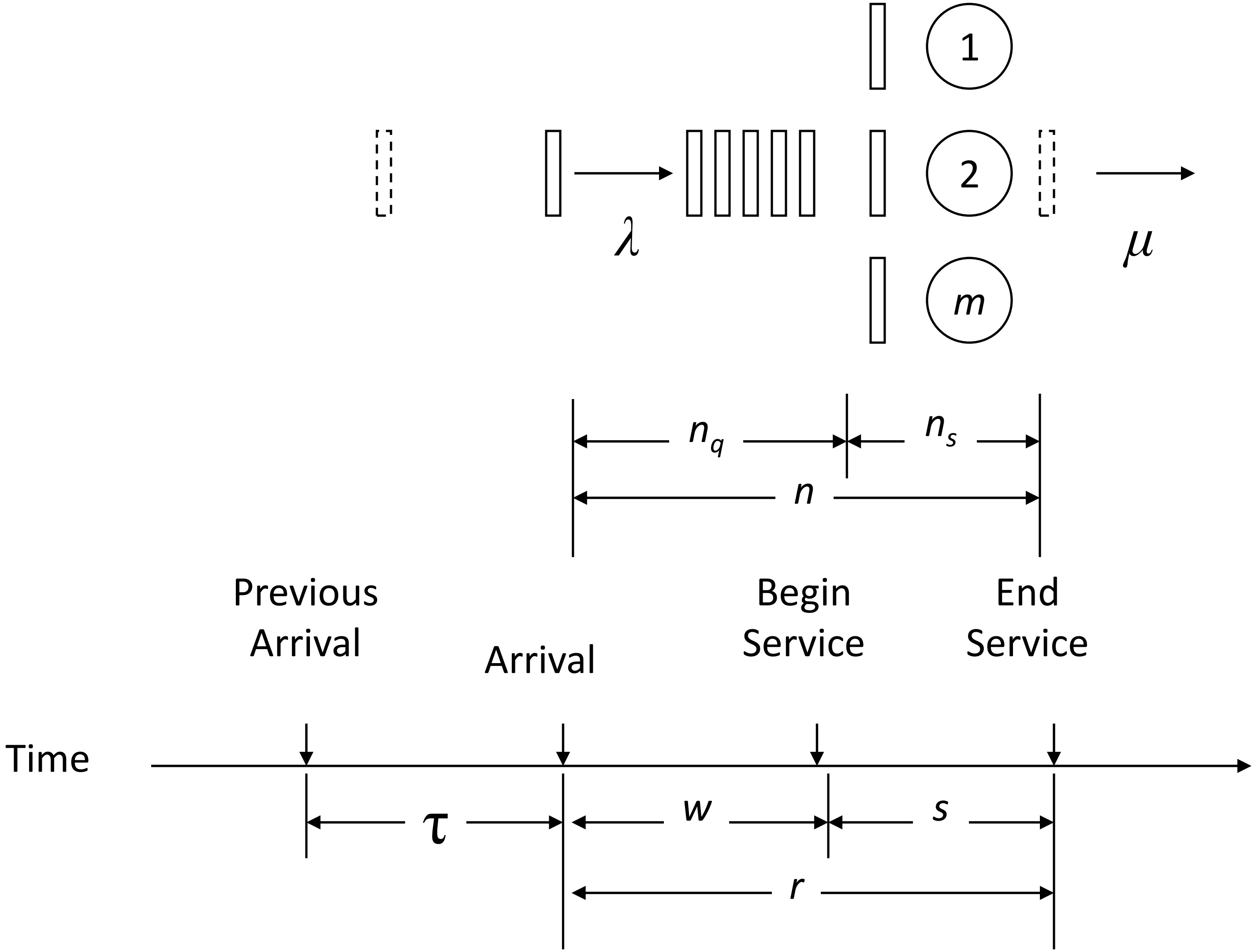
- Time between successive arrivals is exponentially distributed
- Service times are exponentially distributed
- Three servers
- 20 capacity = 3 service + 17 waiting
- If system is full (20), then any arriving jobs are lost (discarded)
- Total of 1500 jobs that can be serviced
- Service discipline is First-Come-First-Served (aka FIFO)

Common Distributions and Assumptions

- ***M*** : Exponential (Markovian, memoryless)
- ***D*** : Deterministic $\Rightarrow$ constant
- ***G*** : General $\Rightarrow$ Any/all distributions

- Default assumptions (unless stated otherwise):
  - Infinite system capacity
  - Infinite population size
  - FCFS service discipline.
- G/G/1 = G/G/1/∞ / ∞ /FCFS

Key Variables (1 of 3)



Key Variables (2 of 3)

- 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 = $1/E[s]$
- n = Number of jobs in the system

Note: Number of jobs in the system includes jobs currently receiving service as well as those waiting in the queue

Key Variables (3 of 3)

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

Outline

1. Basic components of a queue
2. General rules
3. Markov models
4. Common queueing models

Fundamental Rules for All Queues

The following rules apply to all *G/G/m* queues:

1. Stability Condition:

$$\lambda < m\mu$$

- A system is stable if the number of customers waiting in the queue remains finite, or equivalently, the wait time is finite
- Finite-population and finite-capacity systems are always stable

2. Server Utilization: $U = \frac{\lambda}{m\mu}$

3. Mean Number of Busy Servers:

$$E[n_s] = \frac{\lambda}{\mu}$$

Fundamental Rules for All Queues (cont’d)

4. Occupancy:

If jobs are not lost due to insufficient capacity, then:

Mean number of jobs in the system
= Arrival rate × Mean response time

$$\overline{N} = \lambda T$$

Similarly:

Mean number of jobs in the queue
= Arrival rate × Mean waiting time

This is known as Little’s Law (or Conservation Law)

Fundamental Rules for All Queues (cont'd)

5. Number in System versus Number in Queue:

$$n = n_q + n_s$$

Notice that n, n_q, and n_s are random variables

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

6. Time in System versus Time in Queue

$$r = w + s$$

r, w, and s are random variables

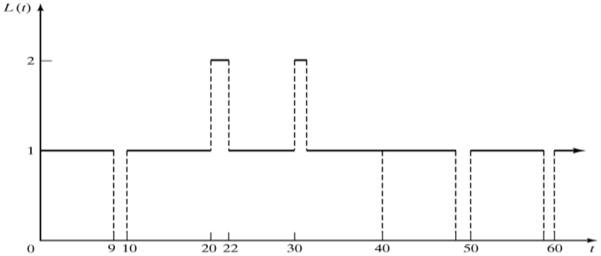
$$E[r] = E[w] + E[s]$$

Example: Effect of Randomness

- A physician who schedules patients every 10 minutes and spends S_i minutes with the i^{th} patient:

$$S_i = \begin{cases} 9 \text{ minutes with probability } 0.9 \\ 12 \text{ minutes with probability } 0.1 \end{cases}$$

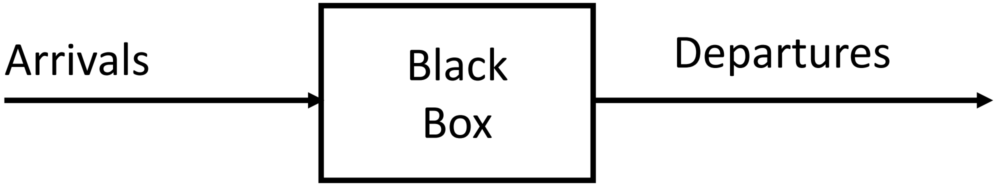
  - Arrivals are deterministic, $\tau_1 = \tau_2 = \ldots = \lambda^{-1} = 10.$
  - Services are stochastic, $E(S_i) = 9.3$ min and $Var(S_i) = 0.9$ min^2.
  - On average, the physician's utilization $= \rho = \frac{\lambda}{\mu} = 0.93 < 1.$
  - Consider the system is simulated with service times: $S_1 = 9, S_2 = 12, S_3 = 9, S_4 = 9, S_5 = 9, \ldots$. The system occupancy becomes:



  - The occurrence of a relatively long service time ($S_2 = 12$) causes a waiting line to form temporarily.

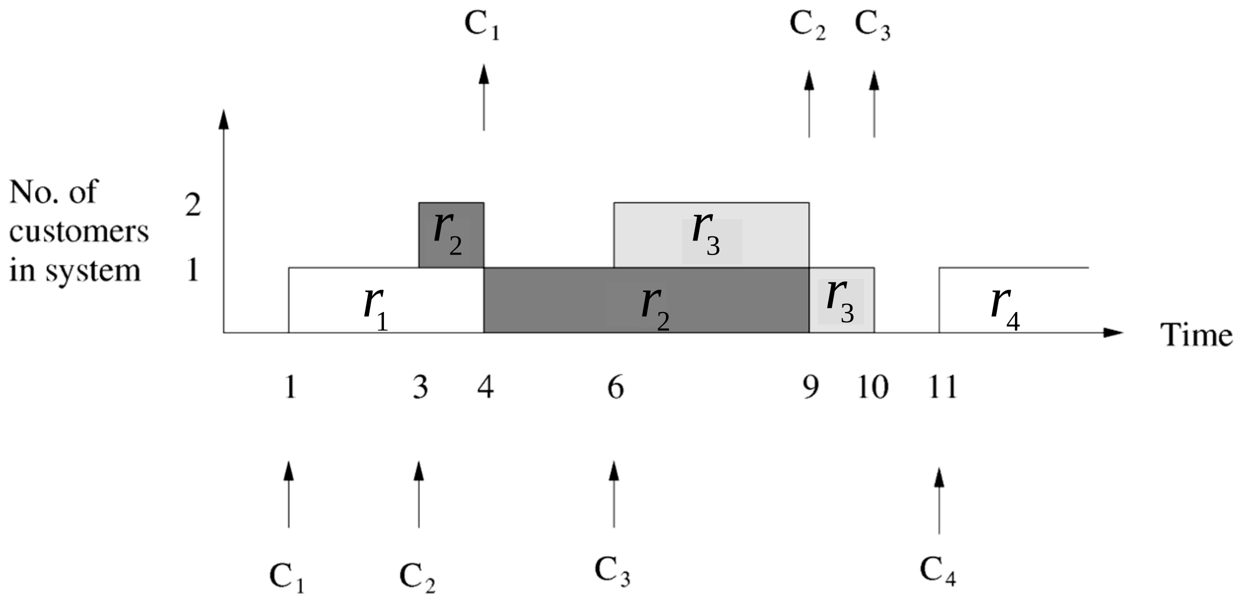
Little's Law

- 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 leaving the system
- Named after John D.C. Little (1961)
- Based on a black-box view of the system:



- In systems in which some jobs are lost due to finite buffers:
  - Use the effective rate of arrivals, e.g., if a portion α are lost, then the effective arrival rate is $(1 - \alpha)\lambda$

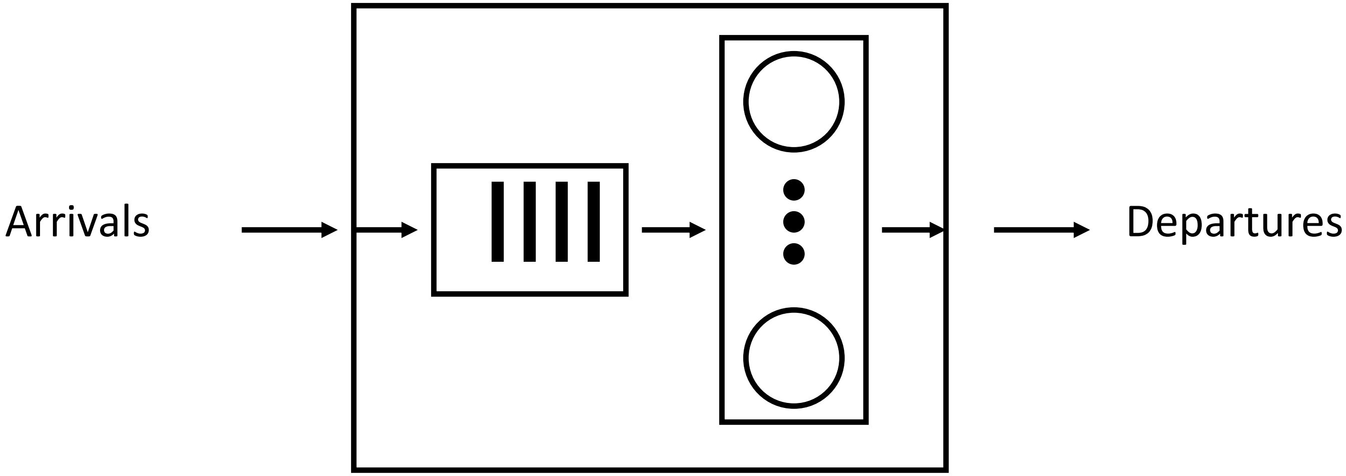
Proof of Little's Law



- Sum of response times is the total area under the curve $= L \cdot E[n] = \sum_{j=1}^{n} r_j =$
- Thus, $\frac{n}{L}\frac{1}{n}\sum_{j=1}^{n} r_j = E[n]$, or $\lambda \cdot E[r] = E[n]$

Little’s Law

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: I/O System Analysis

- 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 at the disk server

  = Arrival rate × Response time

  = 100 (requests/second) ×(0.1 seconds)

  = 10 requests

Some Properties

- Utilization factor, throughput, and Little’s formula are not affected by scheduling discipline

- Little’s formula can also be applied to an arbitrary queueing system (including queueing networks)

Outline

1. Basic components of a queue
2. General rules
3. Markov models
4. Common queueing models

Stochastic Chains

- **Stochastic Process:**
  Collection of random variables indexed over time

- Example: $\{N(t): t \geq 0\}$: number of jobs in system at time t
- Example: $\{W(t): t \geq 0\}$: wait time in system at time t

- State of a process: values it takes over time
  - Discrete: states vary over a finite or countable set
  - Continuous: states vary continuously over a real interval

Stochastic Chain: stochastic process with discrete states

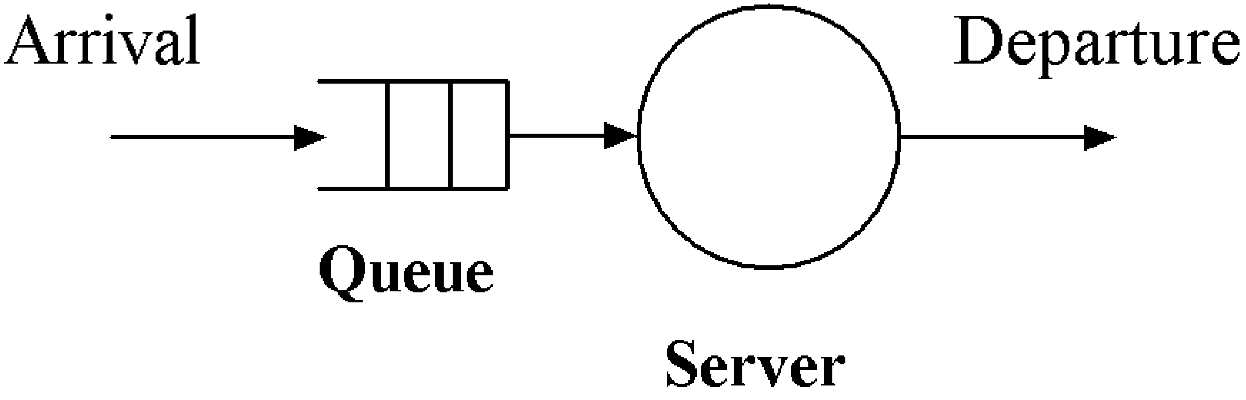
Markov Processes

Named after A. Markov who defined and analyzed them in 1907

- Markov Process: stochastic process in which the future state depends only on the current state, and is independent of the past states of the system
  - It is not necessary to know how long the process has been in the current state to determine the next state
  - State time has a *memoryless (i.e., exponential)* distribution
- ***M/M/m*** queues can be modeled using Markov processes
  - Example: the number of jobs in the queue can be modeled as a Markov chain
    - Markov chain because state space is discrete
    - State = $0, 1, 2, \ldots$

Birth-Death Markov Chain

- Queueing system with a single service facility



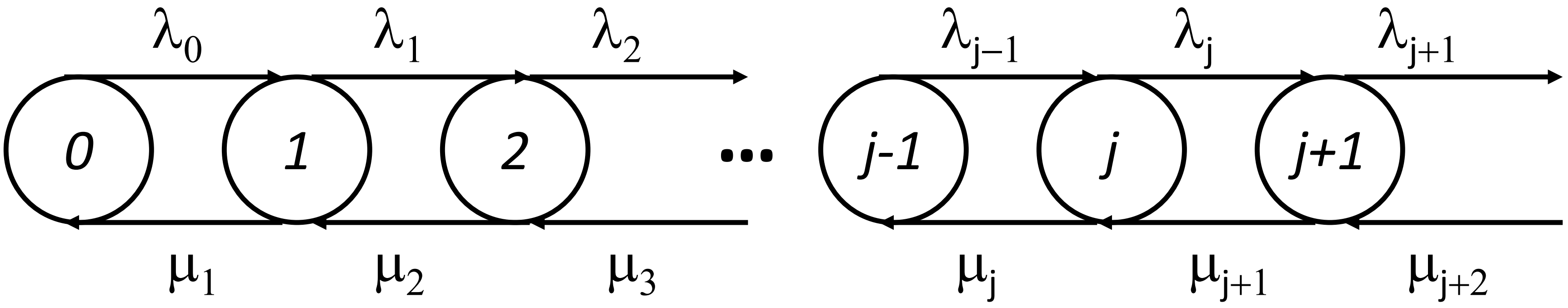
- Interarrival times are *exponentially* distributed
- Service times are *exponentially* distributed
- Arrival rate and service rate may be state dependent, *i.e.*, a function of the state of the system

Birth-Death Markov Chain

- State: Number of jobs n in the system

- Birth: Arrival of a new job changes the system state from n to $n+1$

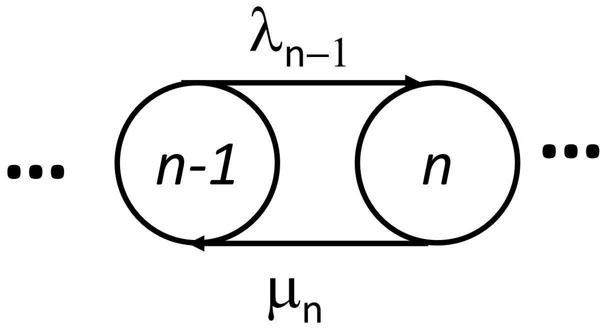
- Death: Departure of a job changes the system state from n to n-1



State-transition diagram

Birth-Death Markov Chain

- When the system is in state n, it has n jobs in it:
  - The new arrivals take place at rate λ_n
  - The service rate is μ_n
- In steady-state: birth rate = death rate
  - If birth rate > death rate: unstable system
  - If birth rate < death rate: empty system



$$\lambda_{n-1} p_{n-1} = \mu_n p_n$$
$$\Rightarrow p_n = \frac{\lambda_{n-1}}{\mu_n} p_{n-1}$$

Steady-State Probability Distribution

- The steady-state probability p_n of a birth-death process being in state n is given by:

$$p_n \quad = \quad \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n} p_0 \quad n = 1, 2, \ldots, \infty$$

- Here, p_0 is the probability of being in the zero state, and can be computed by solving:

$$\sum_{n=0}^{\infty} p_n = 1$$

Normalizing Condition

Basic Solution Steps

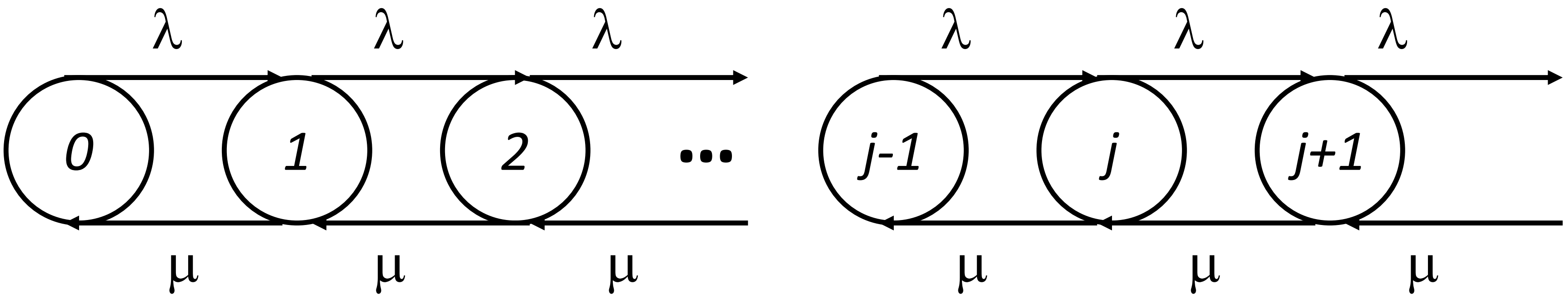
- Compute the steady-state probability distribution. This involves the evaluation of p_0 and p_n
- Compute $E[n] = \text{mean number in system} = \sum_{n=0}^{\infty} np_n$
- Compute $\lambda = \text{mean arrival rate} = \sum_{n=0}^{\infty} \lambda_n p_n$
- Use Little’s Law to obtain $E[r] = \dfrac{E[n]}{\lambda}$

Outline

1. Basic components of a queue
2. General rules
3. Markov models
4. Common queueing models

M/M/1 Queue

- M/M/1 queue is the most commonly used type of queueing model
- Used to model single processor systems or to model individual devices in a computer system
- 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'd)

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

$$p_n = \rho^n p_0$$

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

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

(geometric distribution!)

- *Utilization of the server*

  = Probability of having at least one job in the system

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

Results for M/M/1 Queue (cont'd)

- Mean number of jobs in the system:

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

- Mean number of jobs in the queue:

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

Results for M/M/1 Queue (cont'd)

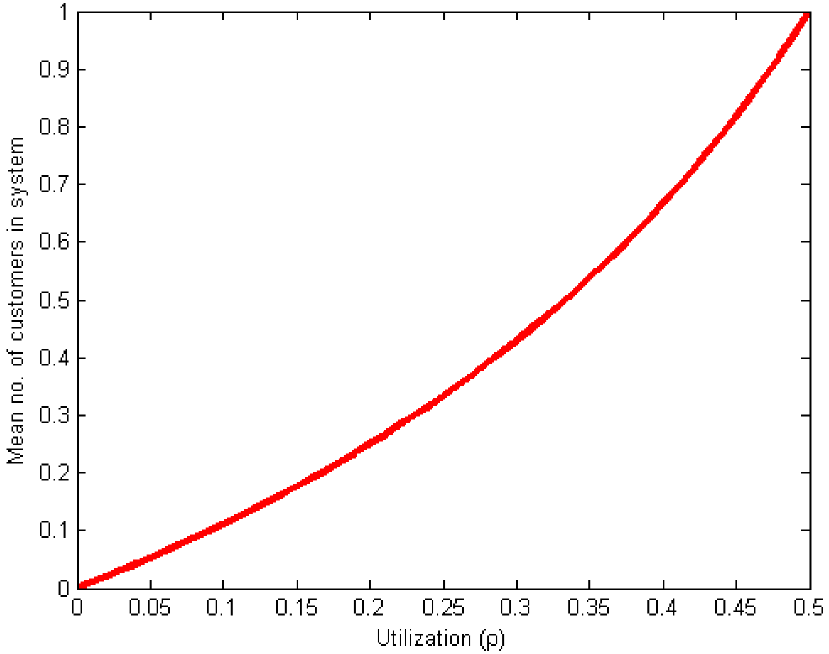
- Probability of *n* or more jobs in the system:

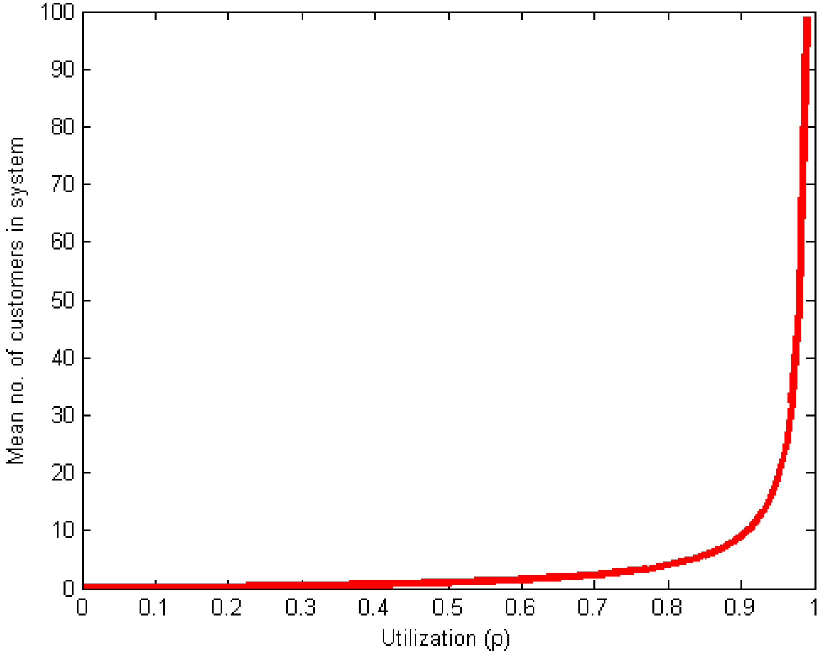
$$P(n \geq k) = \sum_{n=k}^{\infty} p_n = \sum_{n=k}^{\infty} (1-\rho)\rho^n = \rho^k$$

- Mean response time (using Little's Law):
  - Mean number in the system
    = Arrival rate × Mean response time
  - That is: $E[n] = \lambda E[r]$

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

Effect of Server Utilization





- Increase in mean response time and number in system is highly nonlinear as a function of ρ.
- If $\rho \geq 1$, waiting line tends to continually grow in length.

Example: Router Design

- On a network router, measurements show that the packets arrive at a mean rate of 125 packets per second (pps) and the router takes about two milliseconds to forward them. Using an M/M/1 model, analyze the router. What is the probability of buffer overflow if the router had only 11 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
- Router Utilization $\rho = \lambda / \mu = 0.25$
- Probability of n packets in the router = $(1-\rho)\, \rho^n = 0.75(0.25)^n$

Example: Router Design (cont'd)

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

$$
\begin{aligned}
& P(\text{more than 12 packets in the router}) \\
= \quad & \text{P(more than 13 packets in the gateway)} \\
= \quad & \rho^{13} = 0.25^{13} = 1.49 \times 10^{-8} \\
\approx \quad & \text{15 packets per billion packets.}
\end{aligned}
$$

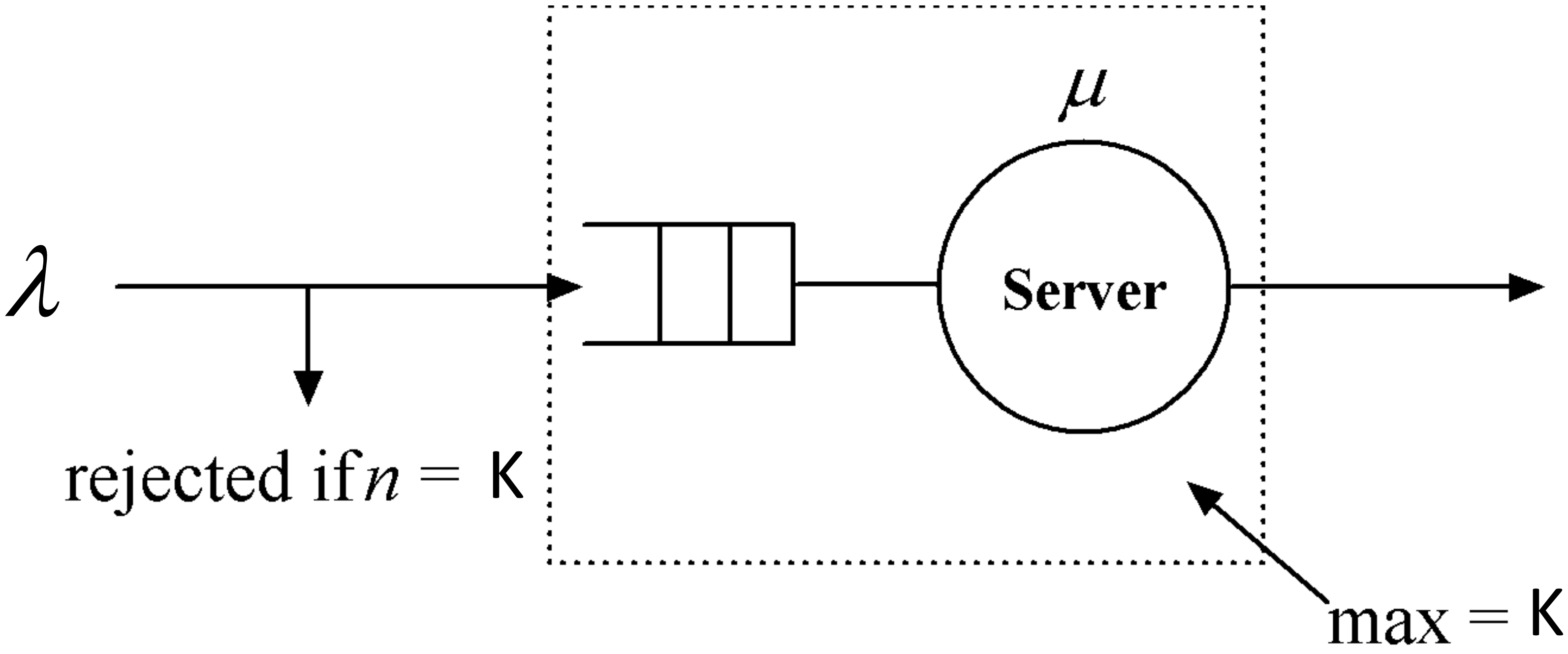
- To limit the probability of loss to less than 10^{-6}:

$$\rho^{(n+1)+1} < 10^{-6} \Rightarrow (n+2)\log(0.25) < -6 \Rightarrow n+2 > 9.97 \Rightarrow n \geq 8$$

Wrong Queuing Model?

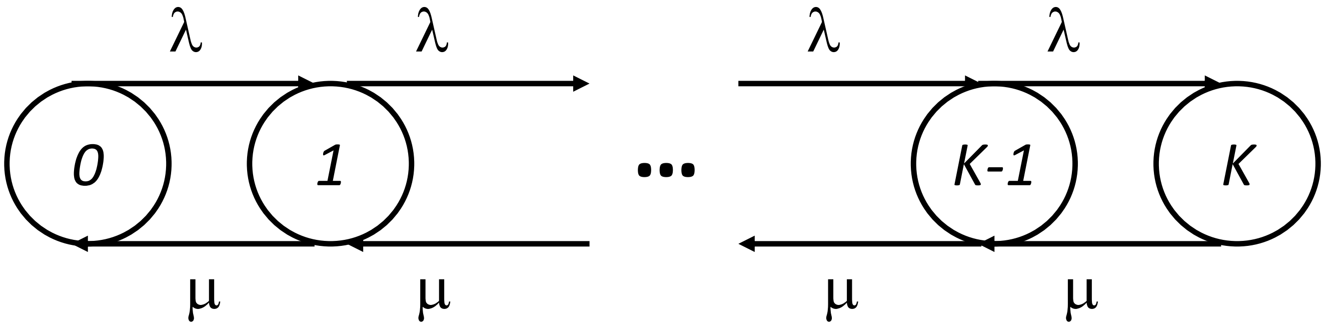
- The last two results about buffer overflow are approximate
- Strictly speaking, the router should actually be modeled as a finite buffer M/M/1/K queue
- Since the utilization is low and the number of buffers is far above the mean queue length, the results obtained are a close approximation

M/M/1/K – Single Server, Finite Queuing Space



Analytic Results

- State-transition diagram:

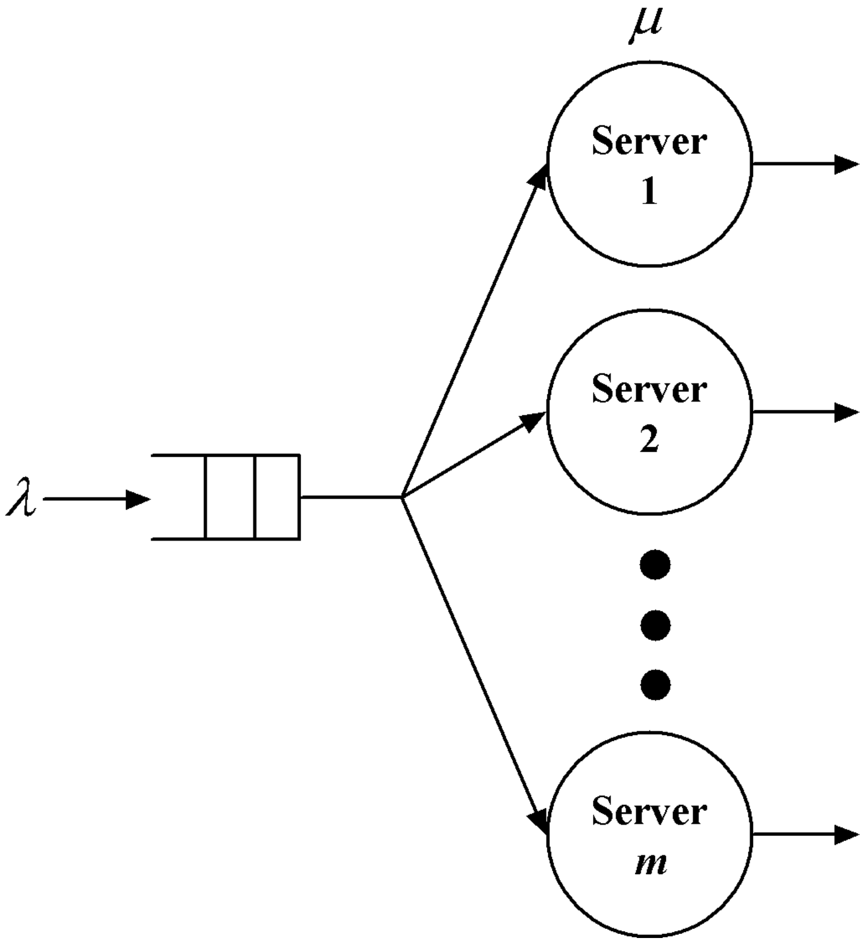


- Solution

$$p_n = p_0 \rho^n, \qquad \text{where } \rho = \frac{\lambda}{\mu}$$

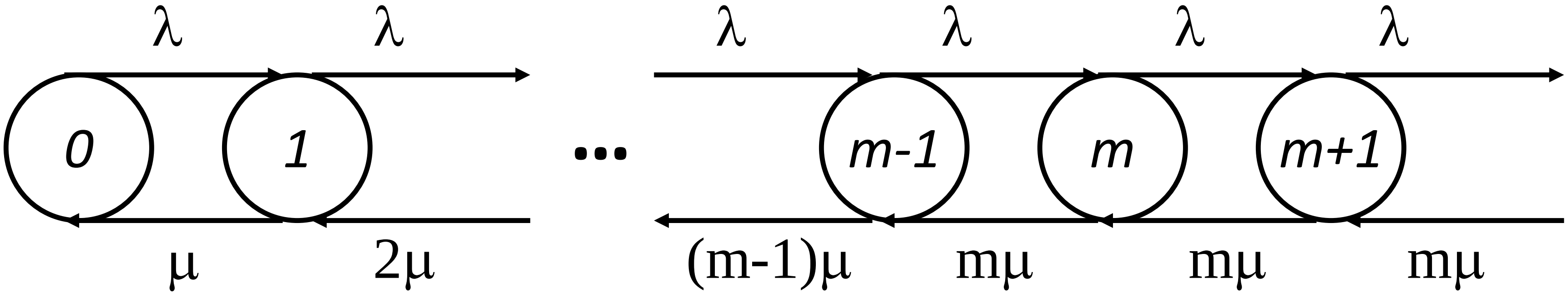
$$p_0 = \left[\sum_{n=0}^{K} \rho^n\right]^{-1} = \begin{cases} \dfrac{1-\rho}{1-\rho^{K+1}} & \rho \neq 1 \\ \dfrac{1}{K+1} & \rho = 1 \end{cases}$$

M/M/m - Multiple Servers



Analytic Results

- State-transition diagram:

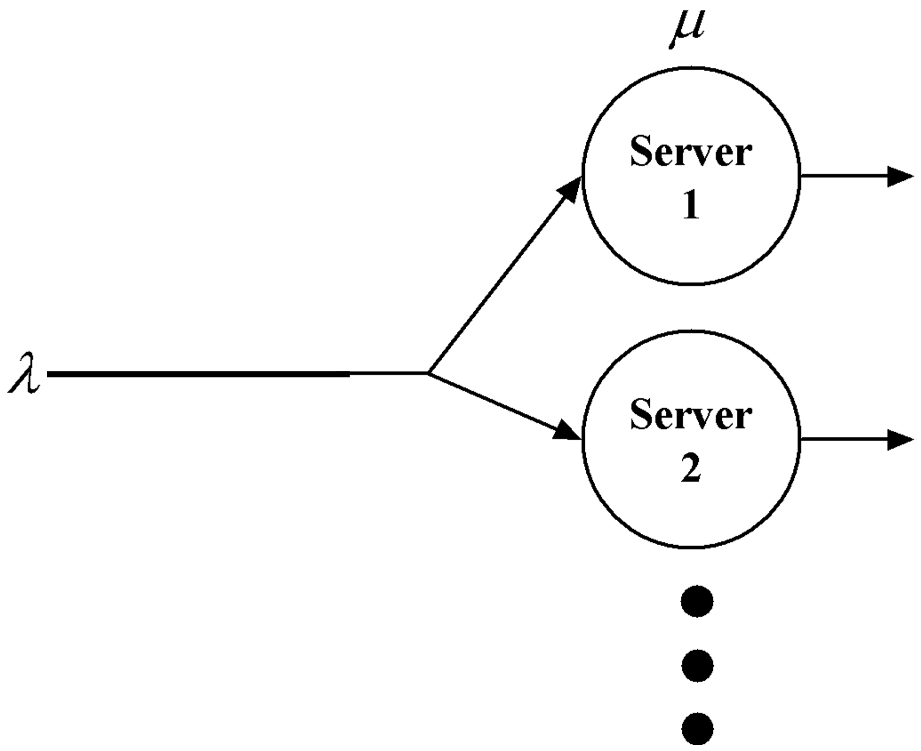


- Solution

$$p_n = p_0 \prod_{j=0}^{n-1} \frac{\lambda_j}{\mu_{j+1}} = \begin{cases} p_0 \rho^n \dfrac{1}{n!} & n \le m \\ p_0 \rho^n \dfrac{1}{m! m^{n-m}} & n > m \end{cases}$$

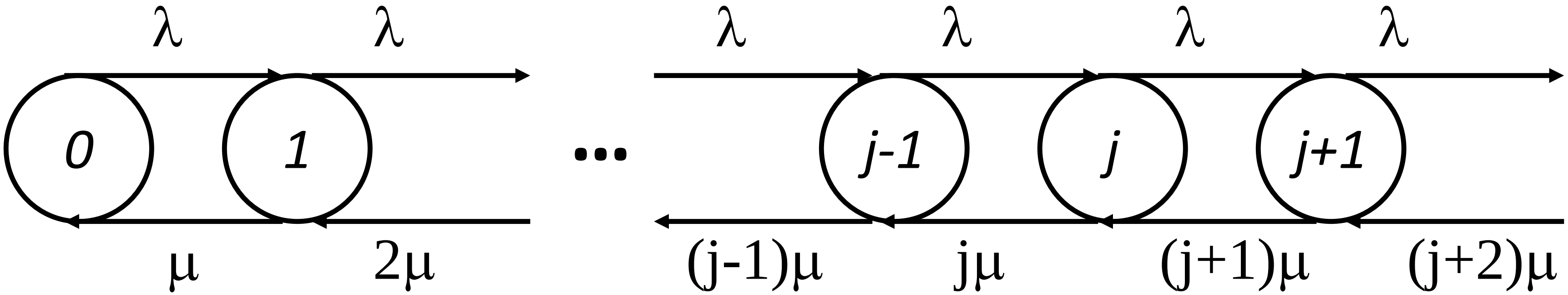
M/M/∞ - Infinite Servers

- Infinite number of servers - no queueing



Analytic Results

- State-transition diagram:



- Solution

$$p_n = p_0 \rho^n \frac{1}{n!}$$

$$p_0 = \left[\sum_{n=0}^{\infty} \rho^n \frac{1}{n!} \right]^{-1} = e^{-\rho}$$

- Thus the number of customers in the system follows a Poisson distribution with rate ρ

UNIVERSITY OF CALGARY

M/G/1 Queue

- Single-server queue with Poisson arrivals, general service time distribution, and unlimited capacity
- Suppose service times have mean $\frac{1}{\mu}$ and variance σ^2
- For $\rho < 1$, the steady-state results for $M/G/1$ are:

$$\boxed{\begin{array}{ll}
\rho = \lambda / \mu, \quad p_0 = 1 - \rho & \\
E[n] = \rho + \dfrac{\rho^2(1+\sigma^2\mu^2)}{2(1-\rho)}, & E[n_q] = \dfrac{\rho^2(1+\sigma^2\mu^2)}{2(1-\rho)} \\
E[r] = \dfrac{1}{\mu} + \dfrac{\lambda(1/\mu^2+\sigma^2)}{2(1-\rho)}, & E[w] = \dfrac{\lambda(1/\mu^2+\sigma^2)}{2(1-\rho)}
\end{array}}$$

M/G/1 Queue

- No simple expression for the steady-state probabilities
- Mean number of customers in service: $\rho = E[n] - E[n_q]$
- Mean number of customers in queue, $E[n_q]$, can be rewritten as:

$$E[n_q] = \frac{\rho^2}{2(1-\rho)} + \frac{\lambda^2 \sigma^2}{2(1-\rho)}$$

  - If λ and μ are held constant, $E[n_q]$ depends on the variability, σ^2, of the service times.

M/G/1 Queue Example

- Example: Two workers are competing for a job. Alex claims to be faster than Bo on average, but Bo claims to be more consistent.
  - Poisson arrivals at rate $\lambda =$ 2 per hour (1/30 per minute).
  - Alex: $\frac{1}{\mu} = 24$ minutes and $\sigma^2 = 20^2 = 400$ minutes^2:

$$E[n_q] = \frac{\left(\frac{1}{30}\right)^2 [24^2 + 400]}{2\left(1 - \frac{4}{5}\right)} = 2.711 \text{ customers}$$

    - The proportion of arrivals who find Alex idle and thus experience no delay at all is $p_0 = 1 - \rho = 1/5 = 20\%$.
  - Bo: $1/\mu = 25$ minutes and $\sigma^2 = 2^2 = 4$ minutes^2:

$$E[n_q] = \frac{\left(\frac{1}{30}\right)^2 [25^2 + 4]}{2\left(1 - \frac{5}{6}\right)} = 2.097 \text{ customers}$$

    - The proportion of arrivals who find Bo idle and thus experience no delay at all is $p_0 = 1 - \rho = 1/6 = 16.7\%$.
  - Although working faster on average, Alex's greater service variability results in an average queue length about 30% greater than Bo's queue.

Effect of Utilization and Service Variability

- For almost all queues, if lines are too long, they can be reduced by decreasing server utilization (ρ) or by decreasing the service time variability (σ^2)

- Coefficient of Variation: a measure of the variability of a distribution

$$CV = \frac{\sqrt{Var(X)}}{E[X]}$$

  - The larger CV is, the more variable is the distribution relative to its expected value.

- Pollaczek-Khinchin (PK) mean value formula:

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

Effect of Utilization and Service Variability

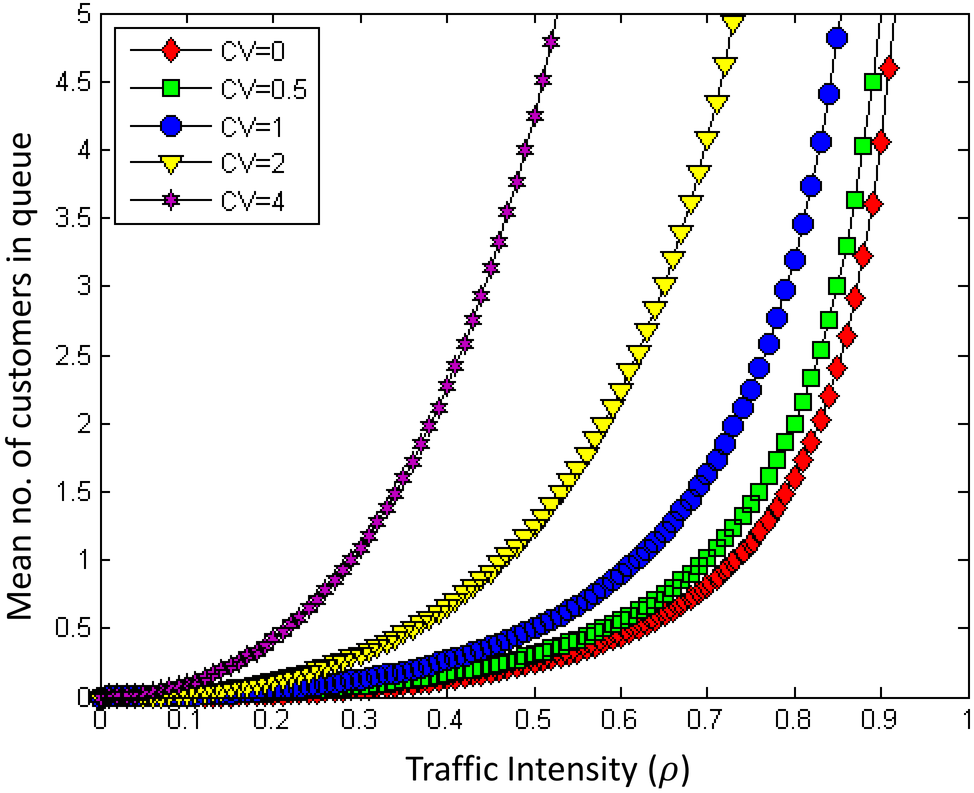
- Consider $E[n_q]$ for *M/G/1* queue:

$$E[n_q] = \frac{\rho^2(1+\sigma^2\mu^2)}{2(1-\rho)}$$

$$= \left(\frac{\rho^2}{1-\rho}\right)\left(\frac{1+(CV)^2}{2}\right)$$

Same as for M/M/1 queue

Adjusts the M/M/1 formula to account for a non-exponential service time distribution


CV=0
CV=0.5
CV=1
CV=2
CV=4
Mean no. of customers in queue
Traffic Intensity (ρ)