

EEC 686/785
Modeling & Performance Evaluation of
Computer Systems

Lecture 20

Wenbing Zhao

Department of Electrical and Computer Engineering
Cleveland State University

wenbing@ieee.org

(based on Dr. Raj Jain's lecture notes)



Outline

- Introduction to queueing theory



Queueing Models

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



Introduction to Queueing Theory

- Queueing notation
- Rules for all queues
- Little's law
- Types of stochastic processes

Basic Components of a Queue

Queueing Notation

- 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
 - SD: service discipline

Arrival Process

- Arrival times: t_1, t_2, \dots, 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 => Poisson
- Erlang
- Hyper-exponential
- General: results valid for all distributions

Service Time Distribution

- Time each student spends at the terminal
- Service times are IID
- Distribution = Exponential, Erlang, Hyper-exponential, General
- Note:
 - Jobs = customers
 - Device = service center = queue
 - Buffer = waiting positions

Service 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)

Service Disciplines

- Infinite Server (IS) = fixed delay
- Shortest Remaining Processing Time first (SRPT)
- Shortest Expected Remaining Processing Time first (SERPT)

Common Distributions

- M : Exponential
- E_k : Erlang with parameter k
- H_k : hyperexponential with parameter k
- D : deterministic => constant
- G : general => all

Exponential Distribution

- Key characteristics
 - Parameters: $1/\lambda = \text{mean}$, $\lambda > 0$
 - Range: $0 \leq x < \infty$
 - pdf: $f(x) = \lambda e^{-\lambda x}$
 - CDF: $F(x) = 1 - e^{-\lambda x}$
 - Mean: $1/\lambda$
 - Variance: $1/\lambda^2$

Exponential Distribution

- **Memoryless property:** past history is not helpful in predicting the future.
 - 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
- **Applications:** to model time between successive events
 - Time between successive request arrivals to a device
 - Time between failures of a device
 - The service times at devices

Poisson Distribution

- To model the number of arrivals over a given interval
- **Key characteristics**
 - Parameters: $\lambda = \text{mean}$, $\lambda > 0$
 - Range: $x = 0, 1, \dots, \infty$
 - pmf: $f(x) = P(X = x) = \lambda^x \frac{e^{-\lambda}}{x!}$
 - Mean: λ
 - Variance: λ^2

Poisson Distribution

- **Applications:** to model the number of arrivals over a given interval
 - Number of requests to a server in a given time interval t
 - Number of component failures per unit time
 - Number of queries to a database system over t seconds
 - Number of typing errors per form
 - Particularly appropriate if the arrivals are from a large number of independent sources

Erlang Distribution

- A random variable x has an Erlang- k ($k = 1, 2, \dots$) distribution with mean k/λ if x is the sum of k independent random variables x_1, \dots, x_k having a common exponential distribution with mean $1/\lambda$
- **Key characteristics**
 - Parameters: $\lambda > 0$, k positive integer
 - Range: $0 \leq x < \infty$
 - pdf: $f(x) = \frac{x^{k-1} e^{-\lambda x}}{(k-1)! / \lambda^k}$ CDF: $F(x) = 1 - e^{-\lambda x} \left[\sum_{i=0}^{k-1} \frac{(\lambda x)^i}{i!} \right]$
 - Mean: k/λ ; Variance: k/λ^2

Erlang Distribution

- Applications: To model service times in a queueing network model
 - A server with Erlang- k service times can be represented as a series of k servers with exponentially distributed service times
 - To model time-to-repair and time-between-failures

Hyper-exponential Distribution

- A random variable x is hyper-exponentially distributed if x is with probability p_i ($i=1, \dots, k$) an exponential random variable x_i with mean λ_i .
- We use the notation $H_k(p_1, \dots, p_k; \lambda_1, \dots, \lambda_k)$, or simply H_k
- pdf: $f(x) = \sum_{i=1}^k p_i \lambda_i e^{-\lambda_i x}$ Mean: $\sum_{i=1}^k p_i / \lambda_i$

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
- \Rightarrow The first three of the six parameters are sufficient, e.g., $G/G/1 = G/G/1/\infty/\infty/FCFS$

Group Arrivals / Service

- Bulk arrivals / service
- $M^{[x]}$: x represents the group size
- $G^{[x]}$ a bulk arrival or service process with general intergroup times
- Examples:
 - $M^{[x]}/M/1$: single server queue with bulk Poisson arrivals and exponential service times
 - $M/G^{[x]}/m$ Poisson arrival process, bulk service with general service time distribution, and m servers

Key Variables

- τ = interarrival 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

Key Variables

- 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
 - Queue length includes jobs currently receiving service as well as those waiting in the queue
- n_q = number of jobs waiting
- n_s = number of jobs receiving service

Key Variables

- r = **response time** or **the time in the system**
= time waiting + time receiving service
- w = waiting time = time between arrival and beginning of service
- All of the above are random variables except λ and μ

Rules for All Queues (G/G/m)

- **Stability condition:** $\lambda < m\mu$ (m : number of servers)
 - Finite-population and the finite-buffer systems are always stable
- **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,

$$\text{Cov}(n_q, n_s) = 0$$

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

Rules for All Queues

- **Number versus time:** if jobs are not lost due to insufficient buffers,

$$\text{Mean number of jobs in the system} = (\text{arrival rate}) \times (\text{mean response time})$$

Similarly

$$\text{Mean number of jobs in the queue} = (\text{arrival rate}) \times (\text{mean waiting time})$$

This is **Little's law**

Rules for All Queues

- **Time in system versus time in queue:**

$$r = w + s$$
 - r , w , and s are random variables

$$E[r] = E[w] + E[s]$$
 - If the service rate is independent of the number in the queue,

$$\text{Cov}(w, s) = 0$$

$$\text{Var}[r] = \text{Var}[w] + \text{Var}[s]$$

Little's Law

- **Mean number of jobs in the system** = (arrival rate) \times (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


```

graph LR
    Arrivals --> BlackBox[Black Box]
    BlackBox --> Departures
          
```
- 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**
 - Once a job enters the queue, it will not be discarded

Proof of Little's Law

- Monitor the system for a time interval T .
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
- From c: Mean time in the system = J/N
- From b: Mean number in the system = $J/T = (N/T) \times (J/N) =$
Arrival rate \times 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 \times mean waiting time
- Similarly, for those currently receiving the service, we have:
Mean number in service
= arrival rate \times 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 \times response time
= (100 requests / second) \times (0.1 seconds)
= 10 requests

Stochastic Processes

- Process**: function of time
- Stochastic process**: random variables, which are functions of time
- Example 1: $n(t)$ = number of jobs at the CPU of a computer system
 - Take several identical systems and observe $n(t)$. The number $n(t)$ is a random variable
 - Can find the probability distribution functions for $n(t)$ at each possible value of t
- Example 2: $w(t)$ = waiting time in a queue

Types of Stochastic Processes

- Discrete or continuous state processes
- Markov processes
- Birth-death processes
- Poisson processes

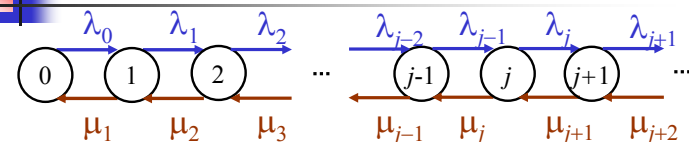
Discrete/Continuous State Processes

- Discrete = finite or countable number of values the state can take
- Number of jobs in a system $n(t) = 0, 1, 2, \dots$
 $n(t)$ is a discrete state process
- The waiting time $w(t)$ is a continuous state process
- **Stochastic chain**: discrete state stochastic process

Markov Processes

- **Markov Processes**: Future states are independent of the past and depend only on the present
- **Markov chain**: discrete state Markov process
- Markov \Rightarrow It is not necessary to know how long the process has been in the current state \Rightarrow state time has a memoryless (exponential) distribution
- M/M/m queues can be modeled using Markov processes
- The time spent by a job in such a queue is a Markov process and the number of jobs in the queue is a Markov chain

Birth-Death Processes



- **Birth-Death Processes**: The discrete space Markov processes in which the transitions are restricted to neighboring states
- Process in state n can change only to state $n+1$ or $n-1$
- Example: the number of jobs in a queue with a single server and individual arrivals (not bulk arrivals)



Poisson Processes

- Interarrival times = IID + exponential
 - Number of arrivals n over a given interval $(t, t+x)$ has a Poisson distribution
 - The arrival process is referred to as a **Poisson process** or a **Poisson stream**