

EEC 686/785  
Modeling & Performance Evaluation of  
Computer Systems

Lecture 23

Wenbing Zhao

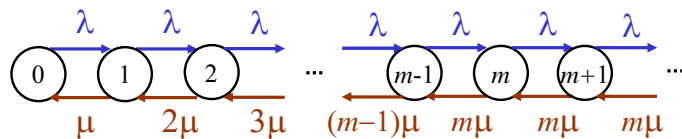
Department of Electrical and Computer Engineering  
Cleveland State University  
wenbing@ieee.org  
(based on Dr. Raj Jain's lecture notes)

Outline

- Review of lecture 22
- Today's topics
  - M/M/m/B queue
  - Queueing networks

M/M/m Queue

- The state of the system is represented by the number of jobs  $n$  in the system



Analysis of M/M/m Queue

- Number of jobs in the system is a birth-death process:

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

$$\mu_n = \begin{cases} n\mu & n = 1, 2, \dots, m-1 \\ m\mu & n = m, m+1, \dots, \infty \end{cases}$$

- Probability of  $n$  jobs in the system:

$$p_n = \begin{cases} \frac{(m\rho)^n}{n!} p_0 & n = 1, 2, \dots, m-1 \\ \frac{\rho^n m^m}{m!} p_0 & n = m, m+1, \dots, \infty \end{cases}$$

## Analysis of M/M/m Queue

- Probability of zero jobs in the system:

$$p_0 = \left[ 1 + \frac{(m\rho)^m}{m!(1-\rho)} + \sum_{n=1}^{m-1} \frac{(m\rho)^n}{n!} \right]^{-1}$$

- Probability that an arriving job has to wait in the queue:

$$\mathcal{G} = P(\geq m \text{ jobs}) = p_0 \frac{(m\rho)^m}{m!(1-\rho)}$$

- This is known as **Erlang's C formula**. Notice that for  $m = 1$ ,  $\mathcal{G} = \rho$

## Analysis of M/M/m Queue

- Mean number of jobs in the queue  $E[n_q]$ :

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

- Expected number of jobs in service:

$$E[n_s] = \sum_{n=1}^{m-1} np_n + \sum_{n=m}^{\infty} mp_n = m\rho$$

- Utilization of each server:

$$U = \frac{\text{Busy time per server}}{\text{Total time}} = \frac{(\lambda T / \mu) / m}{T} = \frac{\lambda}{m\mu}$$

## Analysis of M/M/m Queue

- Mean response time:

$$E[r] = \frac{1}{\mu} \left( 1 + \frac{\mathcal{G}}{m(1-\rho)} \right)$$

- Mean waiting time:

$$E[w] = \frac{\mathcal{G}}{m(1-\rho)}$$

- $q$ -percentile of waiting time:

$$w_q = \max \left\{ 0, \frac{1}{m\mu(1-\rho)} \ln \left( \frac{100\mathcal{G}}{100-q} \right) \right\}$$

## M/M/m/B Queue with Finite Buffers

- Similar to the M/M/m queue but number of buffers  $B$  is finite
- All arrivals, after  $B$  buffers are full, are lost
- $B$  is greater than or equal to  $m$ ; otherwise some servers will never be able to operate => M/M/B/B queue

9

## Analysis of M/M/m/B Queue

- Birth-death process:
 
$$\lambda_n = \lambda \quad n = 0, 1, 2, \dots, B-1$$

$$\mu_n = \begin{cases} n\mu & n = 1, 2, \dots, m-1 \\ m\mu & n = m, m+1, \dots, B \end{cases}$$

11 December 2005      EEC686/785      Wenbing Zhao

10

## Analysis of M/M/m/B Queue

- Probability of  $n$  jobs in the system:
 
$$p_n = \begin{cases} \frac{\lambda^n}{n! \mu^n} p_0 & n = 1, 2, \dots, m-1 \\ \frac{\lambda^n}{m! m^{n-m} \mu^n} p_0 & n = m, m+1, \dots, B \end{cases}$$

In terms of the traffic intensity  $\rho = \lambda/m\mu$ :

$$p_n = \begin{cases} \frac{(m\rho)^n}{n!} p_0 & n = 1, 2, \dots, m-1 \\ \frac{\rho^n m^m}{m!} p_0 & n = m, m+1, \dots, B \end{cases}$$

11 December 2005      EEC686/785      Wenbing Zhao

11

## Analysis of M/M/m/B Queue

- Probability of zero jobs in the system:
 
$$\sum_{n=0}^B p_n = 1$$

$$p_0 + p_0 \sum_{n=1}^{m-1} \frac{(m\rho)^n}{n!} + p_0 \frac{(m\rho)^m}{m!} \sum_{n=m}^B \rho^{n-m} = 1$$

$$p_0 = \left[ 1 + \frac{(1 - \rho^{B-m+1})(m\rho)^m}{m!(1-\rho)} + \sum_{n=1}^{m-1} \frac{(m\rho)^n}{n!} \right]^{-1}$$

11 December 2005      EEC686/785      Wenbing Zhao

12

## Analysis of M/M/m/B Queue

- Mean number of jobs in the system  $E[n]$ :
 
$$E[n] = \sum_{n=1}^B n p_n$$
- Mean number of jobs in the queue  $E[n_q]$ :
 
$$E[n_q] = \sum_{n=m+1}^B (n-m) p_n$$

11 December 2005      EEC686/785      Wenbing Zhao

## Analysis of M/M/m/B Queue

- Variance and other statistics on  $n$  and  $n_q$  can be similarly computed
- All arrivals occurring when the system is in the state  $n=B$  are lost. Rate of the jobs actually entering the system, called **effective arrival rate**, is:

$$\lambda' = \sum_{n=0}^{B-1} \lambda p_n = \lambda \sum_{n=0}^{B-1} p_n = \lambda(1 - p_B)$$

The difference  $\lambda - \lambda' = \lambda p_B$  represents the **loss rate**

## Analysis of M/M/m/B Queue

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

$$E[r] = \frac{E[n]}{\lambda'} = \frac{E[n]}{\lambda(1 - p_B)}$$

Similarly, the mean waiting time is:

$$E[w] = \frac{E[n_q]}{\lambda'} = \frac{E[n_q]}{\lambda(1 - p_B)}$$

## Analysis of M/M/m/B Queue

- Utilization of each server:

$$U = \frac{\text{Busy time per server}}{\text{Total time}} = \frac{(\lambda' T / \mu) / m}{T} = \frac{\lambda'}{m\mu} = \rho(1 - p_B)$$

- Probability of the system in full capacity =  $p_B$

## Analysis of M/M/m/B Queue

- For a M/M/m/m system ( $B=m$ ), loss probability is:

$$p_m = \frac{(m\rho)^m}{m!} p_0 = \frac{(m\rho)^m / m!}{\sum_{j=0}^m \frac{(m\rho)^j}{j!}}$$

This is **Erlang's loss formula**

- Used to compute the probability of lost phone calls.  
Valid also for M/G/m/m queues

## Example 31.5

- Consider the gateway of example 31.1 again. Let us analyze the gateway assuming it has only two buffers. The arrival rate and the service rate, as before, are 125 pps and 500 pps, respectively.
- In this case:  $\lambda=125$ ,  $\mu=500$ ,  $m=1$ , and  $B=2$
- Traffic intensity:  $\rho=\lambda/m\mu=125/(1\times 500)=0.25$

## Example 31.5

- $p_n$  for  $n=1,2,\dots,B$  are:
  - $p_1 = \rho p_0 = 0.25p_0$
  - $p_2 = \rho^2 p_0 = 0.25^2 p_0 = 0.0625p_0$
  - $p_0$  is determined by summing all probabilities:
 
$$p_0 + p_1 + p_2 = 1 \Rightarrow p_0 + 0.25p_0 + 0.0625p_0 = 1$$

$$p_0 = 1 / (1 + 0.25 + 0.0625) = 0.76$$
 Substituting for  $p_0$  in  $p_n$ , we get:
 
$$p_1 = 0.25p_0 = 0.19$$

$$p_2 = 0.0625p_0 = 0.0476$$

## Example 31.5

- Mean number of jobs in the system:
 
$$E[n] = \sum_{n=1}^B n p_n = 1 \times 0.19 + 2 \times 0.0476 = 0.29$$
- Mean number of jobs in the queue:
 
$$E[n_q] = \sum_{n=m}^B (n-m) p_n = (2-1) \times 0.0476 = 0.0476$$
- Effective arrival rate in the system:
 
$$\lambda' = \lambda(1 - p_B) = 125(1 - p_2) = 125(1 - 0.0476) = 119 \text{ pps}$$

## Example 31.5

- Packet loss rate =  $\lambda - \lambda' = 125 - 119 = 6 \text{ pps}$
- Mean response time
 
$$E[r] = \frac{E[n]}{\lambda'} = \frac{0.29}{119} = 2.40 \times 10^{-3} \text{ seconds}$$
- Mean time waiting in the queue
 
$$E[w] = \frac{E[n_q]}{\lambda'} = \frac{0.0476}{119} = 4.0 \times 10^{-4} \text{ seconds}$$

## Example 31.5

- Variance and other statistics for the number of jobs in the system can also be computed since the complete probability mass function  $p_n$  is known. For example:

$$\begin{aligned}\text{Var}[n] &= E[n^2] - (E[n])^2 \\ &= (1^2 \times 0.19 + 2^2 \times 0.0476) - (0.29^2) = 0.2963\end{aligned}$$

- Exercise: What is the probability of buffer overflow?

## Queueing Networks

- Network = model in which jobs departing from one queue arrive at another queue (or possibly the same queue)

## Open Queueing Networks

- Open queueing network: external arrivals & departures
  - > Number of jobs in the system varies with time
  - > Throughput = arrival rate
  - > Goal: to characterize the distribution of number of jobs in the system

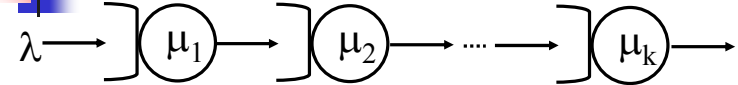
## Closed Queueing Networks

- Closed queueing network: no external arrivals or departures
  - > Total number of jobs in the system is constant
  - > "OUT" is connected back in "IN"
  - > Throughput = flow of jobs in the OUT-to-IN link
  - > Number of jobs is given, determine the throughput

## Mixed Queueing Networks

- Mixed queueing networks: open for some workloads and closed for others
  - Two classes of jobs
  - Class = types of jobs
    - ◆ All jobs of a single class have the same service demands and transition probabilities
    - ◆ Within each class, the jobs are indistinguishable

## Series Networks



- Each individual queue can be analyzed independently of other queues
- Arrival rate =  $\lambda$
- If  $\mu_i$  is the service rate for  $i$ th server:
  - Utilization of  $i$ th server  $\rho_i = \lambda / \mu_i$
  - Probability of  $n_i$  jobs in the  $i$ th queue =  $(1 - \rho_i) \rho_i^{n_i}$   
(result from M/M/1 queue)

## Series Networks

- Joint probability of queue length:

$$\begin{aligned}
 P(n_1, n_2, n_3, \dots, n_M) & \\
 &= (1 - \rho_1) \rho_1^{n_1} (1 - \rho_2) \rho_2^{n_2} (1 - \rho_3) \rho_3^{n_3} \cdots (1 - \rho_M) \rho_M^{n_M} \\
 &= p_1(n_1) p_2(n_2) p_3(n_3) \cdots p_M(n_M)
 \end{aligned}$$

- $\Rightarrow$  Product form network

## Product-Form Network

- Any queueing network in which:

$$P(n_1, n_2, n_3, \dots, n_M) = \frac{1}{G(N)} \prod_{i=1}^M f_i(n_i)$$

When  $f_i(n_i)$  is some function of the number of jobs at the  $i$ th facility,  $G(N)$  is a normalizing constant and is a function of the total number of jobs in the system

## Example 32.2

- Consider a closed system with two queues and  $N$  jobs circulating among the queues:
  - Both servers have an exponentially distributed service time. The mean service times are 2 and 3, respectively

## Example 32.2

- The probability of having  $n_1$  jobs in the first queue and  $n_2 = N - n_1$  jobs in the second queue can be shown to be:

$$P(n_1, n_2) = \frac{1}{3^{N+1} - 2^{N+1}} (2^{n_1} \times 3^{n_2})$$

- The normalizing constant  $G(N)$  is  $3^{N+1} - 2^{N+1}$
- The state probabilities are products of functions of the number of jobs in the queues
- Thus, this is a product form network

## General Open Network of Queues

- Product form networks are easier to analyze
- Jackson (1963) showed that any arbitrary open network of  $m$ -server queues with exponentially distributed service times has a product form

## General Open Network of Queues

- If all queues are single-server queues, the queue length distribution is:

$$\begin{aligned} P(n_1, n_2, n_3, \dots, n_M) &= (1 - \rho_1) \rho_1^{n_1} (1 - \rho_2) \rho_2^{n_2} (1 - \rho_3) \rho_3^{n_3} \cdots (1 - \rho_M) \rho_M^{n_M} \\ &= p_1(n_1) p_2(n_2) p_3(n_3) \cdots p_M(n_M) \end{aligned}$$

Note: queues are not independent M/M/1 queues with a Poisson arrival process

## General Open Network of Queues

- In general, the internal flow in such networks is not Poisson. Particularly, if there is any feedback in the network, so that jobs can return to previously visited service centers, the internal flows are not Poisson

## Closed Product Form Networks

- Gordon and Newell (1967) showed that any arbitrary closed networks of  $m$ -server queues with exponentially distributed service times also have a product form solution

## BCMP Networks

- Baskett, Chandy, Muntz, and Palacios (1975) showed that product form solutions exist for an even broader class of networks.
- This class of networks is called **BCMP Networks**
- BCMP networks satisfy the following conditions:

## BCMP Networks

- **Service disciplines:**
  - First-come-first-served (FCFS)
  - Processor sharing (PS)
  - Infinite servers (IS or delay centers), and
  - Last-come-first-served-preemptive-resume (LCFS-PR)
- **Job classes:**
  - The jobs belong to a single class while awaiting or receiving service at a service center
  - But jobs may change classes and service centers according to fixed probabilities at the completion of a service request

## BCMP Networks

### ■ Service time distributions:

- At FCFS service centers, the service time distributions must be identical and exponential for all classes of jobs
- At other service centers, where the service times should have probability distributions with rational Laplace transforms; different classes of jobs may have different distributions

## BCMP Networks

### ■ State dependent service:

- The service time at a FCFS service center can depend only on the total queue length of the center
- The service time for a class at PS, LCFS-PR, and IS center can also depend on the queue length for that class, but not on the queue length of other classes
- Moreover, the overall service rate of a subnetwork can depend on the total number of jobs in the subnetwork

## BCMP Networks

### ■ Arrival processes:

- In open networks, the time between successive arrivals of a class should be exponentially distributed
- No bulk arrivals are permitted
- The arrival rates may be state dependent
- A network may be open with respect to some classes of jobs and closed with respect to other classes of jobs

## Non-Markovian Product Form Networks

- Denning and Buzen (1978) further extended the product-form networks to non-Markovian networks with the following conditions:
  - Job flow balance
  - One step behavior
  - Device homogeneity

## Non-Markovian Product Form Networks

- **1. Job flow balance:** for each class, the number of arrivals to a device must equal the number of departures from the device
  - The job flow balance assumption holds only in some observation periods
  - However, it is a good approximation for long observation intervals since the ratio of unfinished jobs to completed jobs is small

## Non-Markovian Product Form Networks

- **2. One step behavior:** a state change can result only from single jobs either entering the system, or moving between pairs of devices in the system, or exiting from the system. This assumption asserts that simultaneous job-moves will not be observed
- **3. Device homogeneity:** a device's service rate for a particular class does not depend on the state of the system in any way except for the total device queue length and the designated class's queue length

## The Assumption 3 Implies

- **Single resource possession:** a job may not be present (waiting for service or receiving service) at two or more devices at the same time
- **No blocking:** a device renders service whenever jobs are present; its ability to render service is not controlled by any other device
- **Independent job behavior:** interaction among jobs is limited to queueing for physical devices, for example, there should not be any synchronization requirements

## The Assumption 3 Implies

- **Local information:** a device's service rate depends only on local queue length and not on the state of the rest of the system
- **Fair service:** if service rates differ by class, the service rate for a class depends only on the queue length of that class at the device and not on the queue lengths of other classes
  - This means that the servers do not discriminate against jobs in a class depending on the queue lengths of other classes

## The Assumption 3 Implies

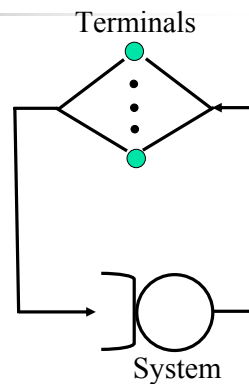
- **Routing homogeneity:** the job routing should be state independent
  - Here, the term routing is used to denote a job's path in the network
  - The routing homogeneity condition implies that the probability of a job going from one device to another device does not depend upon the number of jobs at various devices

## Queueing Network Models of Computer Systems

- Machine repairman model
- Central server model

## Machine Repairman Model

- One of the earliest model of computer systems
- Originally developed for modeling machine repair shops
  - A number of working machines
  - A repair facility with one or more servers (repairmen)
  - Whenever a machine breaks down, it is put in the queue for repair and serviced as soon as a repairman is available



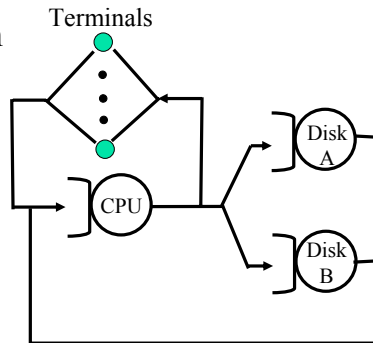
## Machine Repairman Model

- Scherr (1967) used this model to represent a timesharing system with  $n$  terminals
- Users sitting at the terminals generate requests (jobs) that are serviced by the system which serves as a repairman
- After a job is done, it waits at the user-terminal for a random “think-time” interval before cycling again

## Central Server Model

49

- Introduced by Buzen (1973)
- The CPU is the “central server” that schedules visits to other devices
- After service at the I/O devices the jobs return to the CPU



11 December 2005

EEEC686/785

Wenbing Zhao

## Types of Service Centers

50

- **Fixed-capacity service centers:** service time does not depend upon the number of jobs in the device
  - For example, the CPU in a system may be modeled as a fixed-capacity service center
- **Delay centers or IS (infinite server):** no queueing
  - Jobs spend the same amount of time in the device regardless of the number of jobs in it
  - A group of dedicated terminals is usually modeled as a delay center

11 December 2005

EEEC686/785

Wenbing Zhao

## Types of Service Centers

51

- **Load-dependent service centers:** service rates may depend upon the load or the number of jobs in the device
  - M/M/m queue (with  $m \geq 2$ ). Total service rate increases as more and more servers are used
  - A group of parallel links between two nodes in a computer network is another example

11 December 2005

EEEC686/785

Wenbing Zhao