Application of Queuing Models-

Review of Characteristics (calling population system capacity, arrival processes, behavior and disciplines, service times and mechanisms etc) and notations, Application of Long-Run Measures of Performance: Time average in system, average time spent per customer, Little's Formula and server utilization, and costs. Steady State behavior of Infinite (M/G/1, M/M/c/infinity, M/M/c/N/infinity) and finite (M/M/c/K/K) Calling Population Models, Use of Network of Queues.

- Simulation is often used in the analysis of queuing models. In a simple but typical queuing model, shown in Figure 6.1, customers arrive from time to time and join a queue (waiting line), are eventually served, and finally leave the system.
- The term "customer" refers to any type of entity that can be viewed as requesting "service" from a system.
- Therefore, many service facilities, production systems, repair and maintenance facilities, communications and computer systems, and transport and material-handling systems can be viewed as queuing systems.

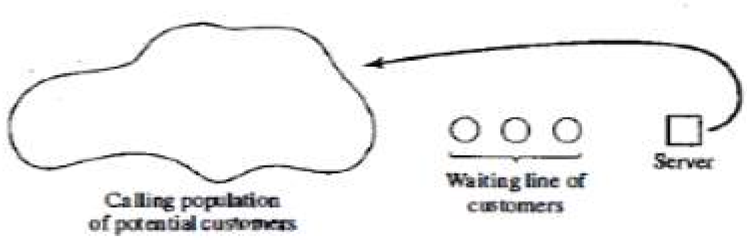


Figure 6.1 Simple queueing model

The Calling Population

- The population of potential customers, referred to as the calling population, may be assumed to be finite or infinite.
- For example, consider a bank of five machines that are curing tires. After an interval of time, a machine automatically opens and must be attended by a worker who removes the tire and puts an uncured tire into the machine.
- The machines are the "customers;' who "arrive" at the instant they automatically open. The worker is the "server," who "serves" an open machine as soon as possible.
- The calling population is finite and consists of the five machines.
- In systems with a large population of potential customers, the calling population is usually assumed to be infinite. For such systems, this assumption is usually innocuous and, furthermore, it might simplify the model.

- Examples of infinite populations include the potential customers of a restaurant, bank, or other similar service facility and also very large groups of machines serviced by a technician.

  System Capacity

  - In any queuing systems, there is a limit to the number of customers that may be in the waiting line or system.
  - For example, an automatic car wash might have room for only 10 cars to waiting the line to enter the mechanism.
  - It might be too dangerous (or illegal) for cars to wait in the street. An arriving customer who finds the system full does not enter but returns immediately to the calling population.
  - Some systems, such as concert ticket sales for students, may be considered as having unlimited capacity, since there are no limits on the number of students allowed to wait 'to purchase tickets.
  - As will be seen later, when a system has limited capacity, a distinction is made between the arrival rate (i.e., the number of arrivals per time unit) and the effective arrival rate (i.e., the number who arrive and enter the system per time unit).

  The Arrival Process

  - The arrival process for infinite-population models is usually characterized in terms of interarrival times of successive customers.
  - Arrivals may occur at scheduled times or at random times. When at random times, the inter arrival times are usually characterized by a probability distribution. In addition, customers may arrive one at a time or in batches.
  - The batch may be of constant size or of random size.
  - One important application of finite population models is the machine-repair problem. The machines are the customers, and a runtime is also called time to failure.
  - When a machine fails, it "arrives" at the queuing system (the repair facility) and remains there until it is "served" (repaired).

  Queue Behavior and Queue Discipline

  - Queue behavior refers to the actions of customers while in a queue waiting for service to begin.
  - In some situations, there is a possibility that incoming customers will balk (leave when they see that the line is too long), renege (leave after being in the line when they see that the line is moving too slowly), or jockey (move from one line to another if they think they have chosen a slow line).

- Queue discipline refers to the logical ordering of customers in a queue and determines which customer will be chosen for service when a server becomes free. OR

- Queue discipline refers to the rule that a server uses to choose the next customer from the queue when the server completes the service of the current customer.

- Common queue disciplines include first-in-first-out (FIFO); last-in-first-out (LIFO); service in random order (SIRO); shortest processing time first (SPT); and service according to priority (PR).

1. First in first out :This principle states that customers are served one at a time and that the customer that has been waiting the longest is served first.
2. Last in first out : This principle also serves customers one at a time, however the customer with the shortest waiting time will be served first. Also known as a stack.
3. Processor sharing: Service capacity is shared equally between customers.
4. Priority : Customers with high priority are served first.[17] Priority queues can be of two types,
5. Non-pre emptive (where a job in service cannot be interrupted) and pre emptive (where a job in service can be interrupted by a higher priority job). No work is lost in either model.
6. Shortest job first: The next job to be served is the one with the smallest size
7. Pre emptive shortest job first:The next job to be served is the one with the original smallest size.
8. Shortest remaining processing time: The next job to serve is the one with the smallest remaining processing requirement.

Service Times and the Service Mechanism

- The service times of successive arrivals are denoted by S_1S1, S_2S2, S_3S3, . . . They may be constant or of random duration.
- In the latter case, {S_1S1, S_2S2, S_3S3, . . . }is usually characterized as a sequence of independent and identically distributed random variables.
- The exponential, Weibull, gamma, lognormal and truncated normal distributions have all been used successfully as models of service times in different situations. Sometimes services are identically distributed for all customers of a given type or class or priority, whereas customers of different types might have completely different service-time distributions.

- In addition, in some systems, service times depend upon the time of day or upon the length of the waiting line. For example, servers might work faster than usual when the waiting line is long, thus effectively reducing the service time.

KENDALL’S NOTATION

Kendall’s Notation is a system of notation according to which the various characteristics of a queuing model are identified.

Kendall (Kendall, 1951) has introduced a set of notations which have become standard in the literature of queuing models. A general queuing system is denoted by (a/b/c): (d/e) where

a	=	probability distribution of the interarrival time.
b	=	probability distribution of the service time.
c	=	number of servers in the system.
d	=	maximum number of customers allowed in the system.
e	=	queue discipline

In addition, the size of the population is important for certain types of queuing problem although not explicitly mentioned in the Kendall’s notation. Traditionally, the exponential distribution

The term system refers to the waiting line plus the service mechanism, but generally it can refer to any sub system of the queue. And the queue refers to the waiting line alone.

Primary Long-Run Measures of Performance of Queuing Systems

- long-run time-average number of customers in system (L)

- long-run time-average number of customers in queue (LQ)
- long-run average time spent in system per customer (W)
- long-run average time spent in queue per customer(W Q)
- server utilization (ρ)

Queuing Models Long-Run Measures of Performance of Queuing Systems Long-Run Measures of Performance of Queuing Systems

This section defines the major measures of performance for a general G /G /c /N /K queuing system, discusses their relationships and shows how they can be estimated from a simulation run. There are two types of estimators:

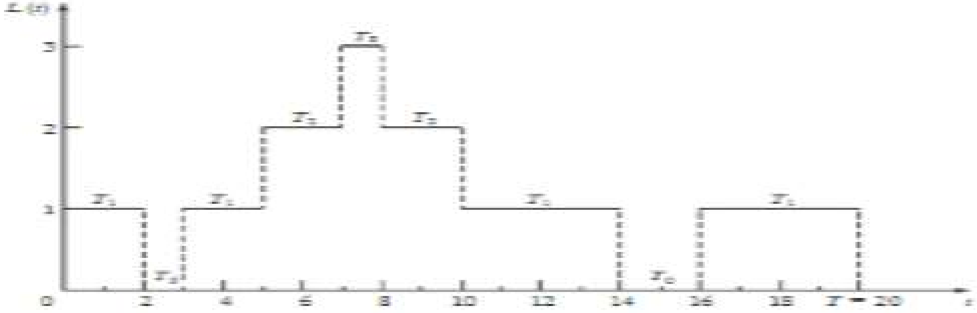
(i) An ordinary sample average

(ii) A time-integrated (time-weighted) sample average.

Time-Average Number in System L

- Consider a queuing system over a period of time T and letL(t) denote the number of customers in the system at time t.
- Let T i denote the total time during [0,T] in which the system contained exactly i customers. The time-weighted-average number in system is defined by

$$\widehat{L} = \frac{1}{T} \sum_{i=0}^{\infty} iT_i = \sum_{i=0}^{\infty} i \left(\frac{T_i}{T} \right)$$



Number in System, L(t) at time t

- It can be seen that the total area under the function L(t) can be decomposed into rectangles of height i and length TiTi .
- It follows that the total area is given by

$$\sum_{i=0}^{\infty} iT_i = \int_0^T L(t)dt,$$ and hence

$$\hat{L} = \frac{1}{T}\sum_{i=0}^{\infty} iT_i = \frac{1}{T}\int_0^T L(t)dt$$

- Time-Average Number in System / Queue L

- If we let L be the long-run time-average number in system,

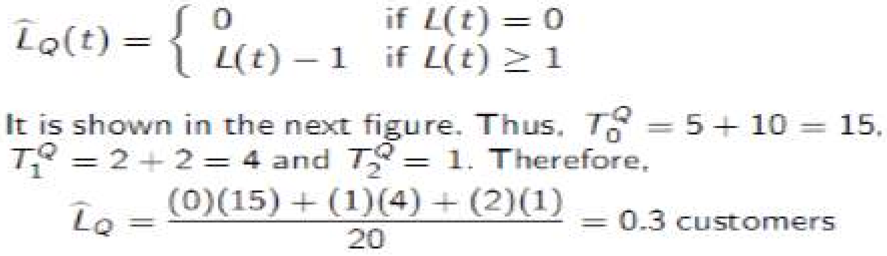
$$\lim_{T\to\infty} \hat{L} = \lim_{T\to\infty} \frac{1}{T}\int_0^T L(t)dt = L$$

The above can be applied to any sub-system of a queuing system. If we let LQ denote the number of customers in line.

$$\hat{L}_Q = \frac{1}{T}\sum_{i=0}^{\infty} iT_i^Q = \frac{1}{T}\int_0^T L_Q(t)dt \to L_Q \text{ as } T \to \infty$$

Time-Average Number in Queue L

- If the previous figure corresponds to a single-server queue- that is a G /G /1/N /K queuing system (N ≥3, K ≥3).
- Then the number of customers waiting in queue is given by LQLQ (t)



$$\widehat{L}_Q(t) = \begin{cases} 0 & \text{if } L(t) = 0 \\ L(t) - 1 & \text{if } L(t) \geq 1 \end{cases}$$

It is shown in the next figure. Thus, $T_0^Q = 5 + 10 = 15$, $T_1^Q = 2 + 2 = 4$ and $T_2^Q = 1$. Therefore,

$$\widehat{L}_Q = \frac{(0)(15) + (1)(4) + (2)(1)}{20} = 0.3 \text{ customers}$$

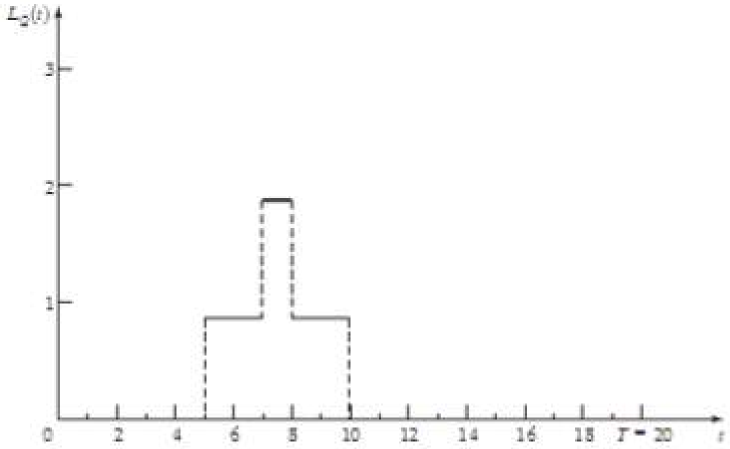
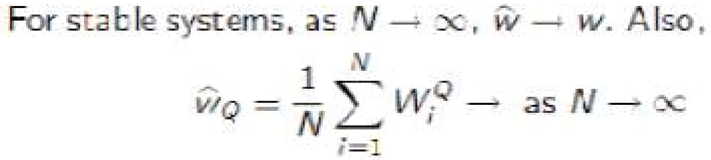


Figure: Number in Queue, $L_Q(t)$ at time t

Average Time Spent in System per Customer W

- If W1W1, W2W2,...,WnWn nare the times each customer spends in system during [0,T], whereN is the number of arrivals during that time period, the average time spent in system per customer (average system time) is

$$\widehat{w} = \frac{1}{N}\sum_{i=1}^{N} W_i$$



For stable systems, as $N \rightarrow \infty$, $\widehat{w} \rightarrow w$. Also,

$$\widehat{w}_Q = \frac{1}{N}\sum_{i=1}^{N} W_i^Q \rightarrow \text{ as } N \rightarrow \infty$$

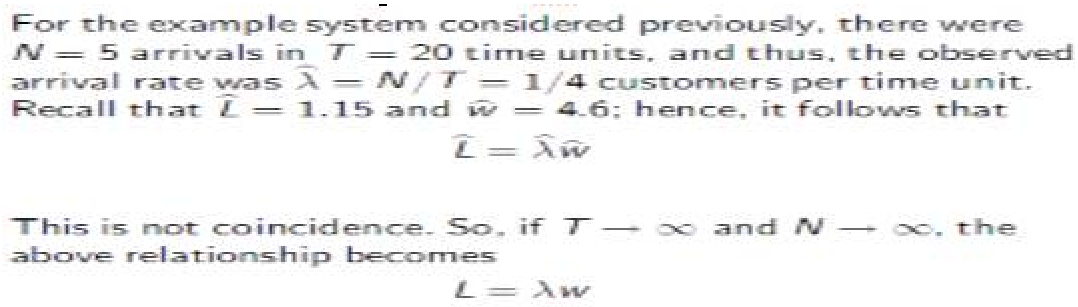
- For stable systems, as

$N \rightarrow \infty$, $\widehat{w} \rightarrow w$

with probability 1,where w is called the long-run average system time. Also,

$$\widehat{w}_Q = \frac{1}{N}\sum_{i=1}^{N} W_i^Q \rightarrow \text{ as } N \rightarrow \infty$$

- The Conservation Equation L= λW



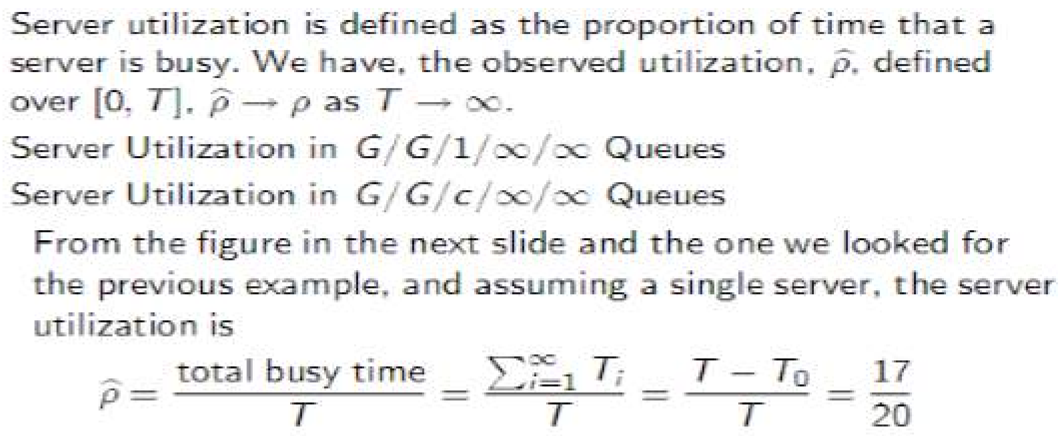
For the example system considered previously, there were $N = 5$ arrivals in $T = 20$ time units, and thus, the observed arrival rate was $\widehat{\lambda} = N/T = 1/4$ customers per time unit. Recall that $\widehat{L} = 1.15$ and $\widehat{w} = 4.6$; hence, it follows that

$$\widehat{L} = \widehat{\lambda}\widehat{w}$$

This is not coincidence. So, if $T \rightarrow \infty$ and $N \rightarrow \infty$, the above relationship becomes

$$L = \lambda w$$

- Server Utilization



Server utilization is defined as the proportion of time that a server is busy. We have, the observed utilization, $\widehat{\rho}$, defined over $[0, T]$, $\widehat{\rho} \rightarrow \rho$ as $T \rightarrow \infty$.

Server Utilization in $G/G/1/\infty/\infty$ Queues

Server Utilization in $G/G/c/\infty/\infty$ Queues

From the figure in the next slide and the one we looked for the previous example, and assuming a single server, the server utilization is

$$\widehat{\rho} = \frac{\text{total busy time}}{T} = \frac{\sum_{i=1}^{\infty} T_i}{T} = \frac{T - T_0}{T} = \frac{17}{20}$$

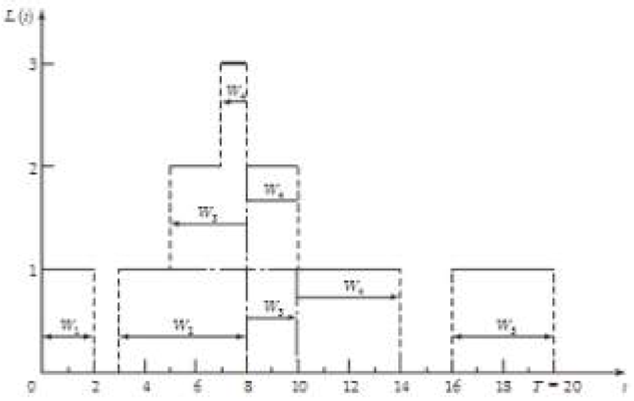


Figure: System Times, W_i for Single-Server FIFO Queuing System

Use of Network of Queues.

Network Queueing

Network queueing is a very important application of queueing theory. The term 'network of queues' describes a situation where the input from one queue is the output from one or more others. This is true in many situations from telecommunications to a PC. Below is a description of some of the broad applications of network queueing describing how theories apply to them. Also there are a few general network queueing theories.

• Computer Networks

A simple example of network queueing is the central server network. This consists of a CPU (Central Processing Unit), storage units it can access and input devices to access it. The task the CPU performs are queued on different criteria. Also, the storage units could have their own individual queues. Queues tend to be ordered in a number of ways. They can also be executed either on a one by one serial basis or bit by bit by Time Sharing. It is not always neccessary to treat customers in a queue equally. A priority queueing system may often be used to give some jobs preferential treatment.

- **Time Sharing**

  Time sharing is when the CPU is dedicated for one task for a fixed period of time after which it is switched to another task. The task can then be recycled ie put back in the queue so that the remainder of it is executed at another time. This can be repeated until the task is complete.

- **Orders for job queues**

  A job queue can be executed one by one or alternatively via time sharing but either way when the jobs are in the queue, the order in which they are executed is crucial.

  First in First Out (FIFO)
  A first in first out queue is the same basis we would (hopefully) use in real life for a cinema queue or a phone box in that the tasks are executed in the order that they arrived.

  Last in First Out (LIFO)
  A last in first out queue is precisely the opposite as a new job is started as soon as it is queued. This may not be as foolish as it sounds as if the job is done on a time sharing basis it may be sensible to start it as soon as possible then switch back to other jobs for a while. This method may imply the need for pre-emption ie the ability to stop a job half way through to start another

one. On the other hand the job could be finished before the new one is started or perhaps the queue does not need the feature of being able to add a job half way through.

Smallest job first (SJF)

A smallest job first queue orders the jobs in terms of the smallest one first which means you get as many jobs complete as quickly as possible even though in total the time taken is the same.

- **Priority Queues**

  An example of a priority queue is a PC. There is a queue of events recieved from each input device(ie mouse, keyboard etc). Imagine a system with two queues, one for the mouse and one for the keyboard which lead into a master queue of input events. Mouse movement may be given priority over mouse button presses. Any event from the mouse may be given priority in the master queue over any event in the keyboard queue.

- **Emption and Pre-Emption**

  It is important in this system to have a way of ealing with a situation where a customer of priority 1 arrives in a queue headed by a customer of priority 2 (ie 1 has higher priority than 2). This situation can be dealt with either in an emptive or pre-emptive fashion. In an emptive system the new entry waits for the other to be completed before beginning. In a pre-emptive system the queue can stop the current entry half way through it's execution to start the new one.

• Network communication

There are several broad methods connected to network communication:

- **Circuit Switching**

  When a call is made from a source to a destination it must traverse several nodes along the way. Which nodes it traverses is determined by the availability of free channels along the way. Each node has a queue for calls requesting a channel. Once a channel has been opened the call can progress to the next node and wait for a channel there. The channel remains open until the source or destination (once reached) closes the call.

- **Packet Switching**

  Messages are transmitted through intermediate stages and the route a message takes depends entirely upon the current load on the system. The route allocation is dynamic. Each stage requires a random amount of time reflecting the length of the queue at that stage.

• Broadcasting

- **Radio Communication**

  Considering the nodes as transmitters/recievers you can treat each as having a queue for their channels. Without going into great detail of the various systems used: it is always neccessary to consider the fact that a to open a channel you must check to see if the two adjacent channels are also free as interference blocks transmissions. When the channels are not free it may be neccessary to re-allocate communications that already have channels to make room.

- **Digital Communication**

  This is done on the basis of time slots. For a given communication link it could have several or all slots filled and no interference would take place making allocation far simpler. The aspect of nodes with queues still applies however.

• Markov Chains

A stochastic process is a Markov process if the future of the process depends only upon the present state of the process implying that the present state is a direct result of its history. A Markov process is called a Markov chain if its state space is discrete ie it moves from one clearly defined state to another.

• The Jackson Queueing Network

Consider a FIFO queueing network: A queueing network is closed if there are no arrivals from outside of it. Clearly the nature of this system will be that of Markovian Chain as there are only a finite number of states of the network and the next state depends on the ones previous to it. However, a network can also be open. An open queueing network can recieve customers from outside to recieve services at its nodes. Jackson was the first to consider such systems in detail. It is assumed that the arrival rate of customers conforms to the poisson distribution. The average service time at each node and the probability of a node leaving the network is known. From this Jackson formulated a way of calculating the Markovian equilibrium state of the network. ie a tuple containing the equilibrium number of customers at every node.

• Other Network types

The Gordon-Newell network is essentially a closed version of the Jackson network while the BCMP (Baskett, Chandy, Muntz and Palacios) network includes facility for different classes of customers. This requires indexing of the different service time requirements at each node plus different leaving probabilities for each class. Variations of this network are used commonly in computer systems and networks today.