Simulation involves the development of descriptive computer models of a system and exercising those models to predict the operational performance of the underlying system being modeled. Systems that change with time, such as a gas station where cars come and go (called dynamic systems) and involve randomness. Nobody can guess at exactly which time the next car should arrive at the station, are good candidates for simulation. Modeling complex dynamic systems theoretically need too many simplifications and the emerging models may not be therefore valid.

Suppose we are interested in a gas station. We may describe the behavior of this system graphically by plotting the number of cars in the station; the state of the system. Every time a car arrives the graph increases by one unit while a departing car causes the graph to drop one unit. This graph (called sample path), could be obtained from observation of a real station, but could also be artificially constructed. Such artificial construction and the analysis of the resulting sample path (or more sample paths in more complex cases) consist of the simulation.

Time Graph Representation

Every System based on the change of time. So in a system model time counting is a crucial thing. In a graph time is recorded by a number called clock time or time counter. Initially it is set on zero. Two basic methods exits for updating clock time.

1. **Time Slicing:** Advances the model by a fixed amount each time, regardless of the absence of any events to carry out.
2. **Next Event:** Advances the model to the next event to be executed, regardless of the time interval. This method is more efficient than Time Slicing, especially where events are infrequent, but can be confusing when being represented graphically (processes that take different times will appear to happen in the same time frame if the stop event is the next event after the start event).

The first method is called "Interval-oriented" and another one is called "Event-oriented". Since the early 1960's, Simulation has been one of many methods used to aid strategic decision-making within industry. Its main strength lies in the ability to imitate complex real world problems and to analyze the behaviour of the system as time progresses.

Simulation is often used in the analysis of queuing models. In a simple typical queuing model, shown in fig 1, customers arrive from time to time and join a queue or waiting line, are eventually served, and finally leave the system.

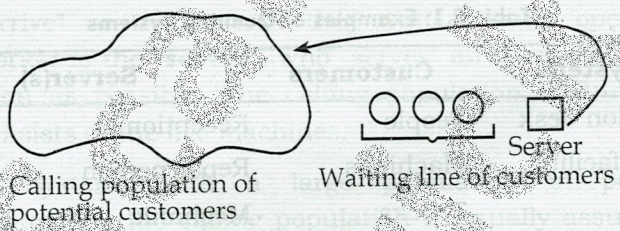

Fig 3.1: Simple Queuing Model

The term "customer" refers to any type of entity that can be viewed as requesting "service" from a system. Queuing models, whether solved mathematically or analyzed through simulation, provide the analyst with a powerful tool for designing and evaluating the performance of queuing systems. Typical measure of performance include server utilization, length of waiting lines and delays of customers. A queue is waiting line for service i.e the combination of all entities in the system-those being served, and those waiting for service-will be called a queue. People, cars, trucks, ships, T.V's, arrive at a certain place to be serviced in some way. The important parameters in a queuing system are:

1. The arrival pattern of customers
2. The service pattern
3. The no. of servers
4. The queue discipline

CHARACTERISTICS OF QUEUING SYSTEMS

The key elements, of a queuing system are the customers and servers. The term "**customer**" can refer to people, machines, trucks, mechanics, patients—anything that arrives at a facility and requires service. The term "server" might refer to receptionists, repairpersons, CPUs in a computer, or washing machines.... any resource (person, machine, etc. which provides the requested service. The following Table 3.1 lists a number of different queuing systems.

Table 3.1: Examples of Queuing Systems

System	Customers	Server(s)
Reception desk	People	Receptionist
Repair facility	Machines	Repairperson
Garage	Trucks	Mechanic
Tool crib	Mechanics	Tool-crib clerk
Hospital	Patients	Nurses
Warehouse	Pallets	Crane
Airport	Airplanes	Runway
Production line	Cases	Case packer
Warehouse	Orders	Order picker
Road network	Cars	Traffic light
Grocery	Shoppers	Checkout station
Laundry	Dirty linen	Washing machines/dryers
Job shop	Jobs	Machines/workers
Lumberyard	Trucks	Overhead crane
Saw mill	Logs	Saws
Computer	Jobs	CPU, disk, tapes
Telephone	Calls	Exchange
Ticket office	Football fans	Clerk
Mass transit	Riders	Buses, trains

The elements of a queuing system are:-

1. 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 5 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 finite or infinite. Examples of infinite populations include the potential customers of a restaurant, bank, etc.

The main difference between finite and infinite population models is how the arrival rate is defined. In an infinite-population model, the arrival rate is not affected by the number of customers who have left the calling population and joined the queuing system. On the other hand, for finite calling population models, the arrival rate to the queuing system does depend on the number of customers being served and waiting.

2. System Capacity

In many 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 may have room for only 10 cars to wait in line to enter the mechanism.

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. There are no limits on the number of students allowed to wait to purchase tickets.

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).

3. The Arrival Process

Arrival process for infinite-population models is usually characterized in terms of inter arrival 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.

The most important model for random arrivals is the Poisson arrival process. If A_n represents the inter arrival time between customer $n-1$ and customer n (A_1 is the actual arrival time of the first customer), then for a Poisson arrival process, A_n is exponentially distributed with mean $1/\lambda$ time Units. The arrival rate is λ customers per time unit. The number of arrivals in a time interval of length t, say $N(t)$, has the Poisson distribution with mean λ_t *customers.*

The Poisson arrival process has been successfully employed as a model of the arrival of people to restaurants, drive-in banks, and other service facilities.

A second important class of arrivals is the scheduled arrivals, such as patients to a physician's office or scheduled airline flight arrivals to an airport. In this case, the inter arrival times $\{A_n, n = 1,2,\ldots\}$ may be constant, or constant plus or minus a small random amount to represent early or late arrivals.

A third situation occurs when at least one customer is assumed to always be present in the queue, so that the server is never idle because of a lack of customers. For example, the "customers" may represent raw material *for a* product, and sufficient raw material is assumed to be always available.

For finite-population models, the arrival process is characterized in a completely different fashion. Define a customer as *pending* when that customer is outside the queuing system and a member of the potential calling population.

Runtime of a given customer is defined as the length of time from departure from the queuing system until that customer's next arrival to the queue.

Let A_i, A_2,... be the successive runtimes of customer /, and let S_1, S_2..... be the corresponding successive system times; that is, $S^{(i)}_n$ is the total time spent in the system by customer i during the nth visit. Figure 2 illustrates these concepts for machine 3 in the tire-curing example. The total arrival process is the superposition of the arrival times of all customers fig 3.2 shows the first and second arrival of machine 3.

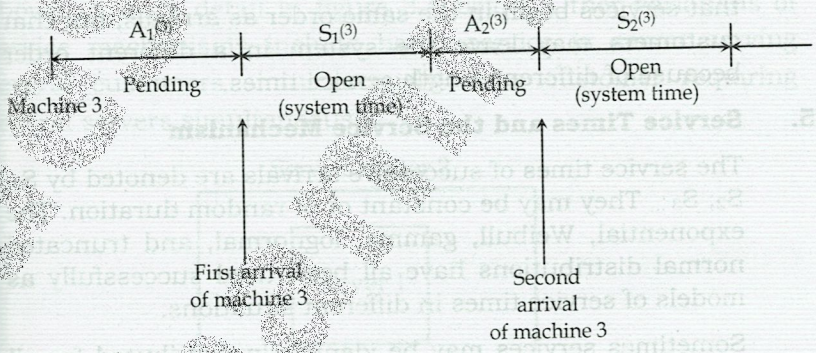

Fig 3.2: Arrival process for a finite-population model.

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" (r pair d). Tim s to failure for a given class of machine have been characterized by the exponential, the Weibull, and the gamma distributions. Models with an exponential runtime are sometimes analytically tractable.

4. Queue Behavior and Queue Discipline

Queue behavior refers to customer actions while in a queue waiting for service to begin. In some situations, there is a possibility that incoming customers may 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.

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).

In a job shop, queue disciplines are sometimes based on due dates and on expected processing time for a given i type of job. Notice that a FIFO queue discipline implies that services begin in the same order as arrivals, but that customers may leave the system in a different order because of different-length service times.

5. Service Times and the Service Mechanism

The service times of successive arrivals are denoted by S_1, S_2, S_3 ... They may be constant or of random duration. The exponential, Weibull, gamma, lognormal, and truncated normal distributions have all been used successfully as models of service times in different situations.

Sometimes services may be identically distributed for all customers of a given type or class or priority, while customers of different types may have completely different service-time distributions. In addition, in some systems, service times depend upon the time of day or the length of the waiting line. For example, servers may work faster than usual when the waiting line is long, thus effectively reducing the service times.

A queuing system consists of a number of service centers and interconnecting queues. Each service center consists of some number of servers, *c,* working in parallel; that is, upon getting to the head of the line, a customer takes the

first available server. Parallel service mechanisms are either single server ($c = 1$), multiple server ($1 < c < \infty$), or unlimited servers ($c = \infty$). (A self-service facility is usually characterized as having an unlimited number of servers.)

Example : *Consider a discount warehouse where customers may either serve themselves; or w t f* of three clerks, and finally leave after paying a single cashier. The system is represented by the flow diagram in figure 3.3 below:

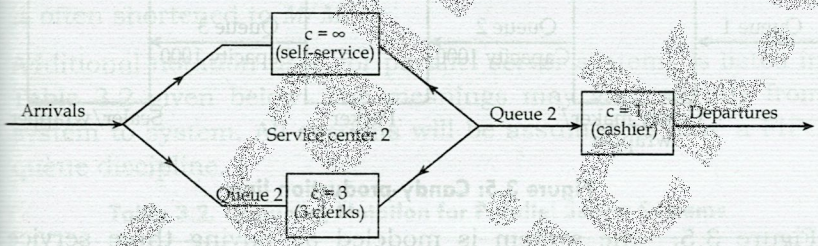

Figure 3.3: Discount warehouse with three service centers

The subsystem, consisting of queue 2 and service center 2, is shown in more detail in figure 3.4 below. Other variations of service mechanisms include batch service (a server serving several customers simultaneously) or a customer requiring several servers simultaneously.

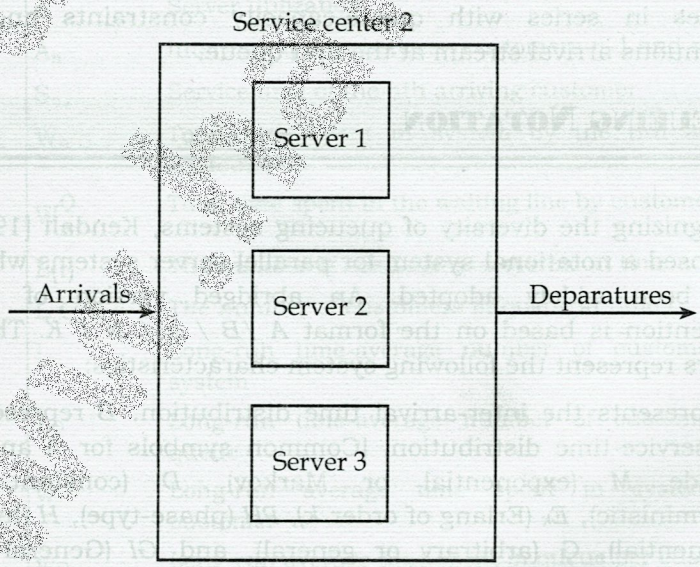

Figure 3.4: Service center 2, with c = 3 parallel servers

Example: A candy manufacturer has a production line which consists of three machines separated by inventory-in-process buffers. The first machine makes and wraps the individual pieces of candy, the second packs 50 pieces in a box, and the third seals and wraps the box. The two inventory buffers have capacities of 1000 boxes each. As illustrated by

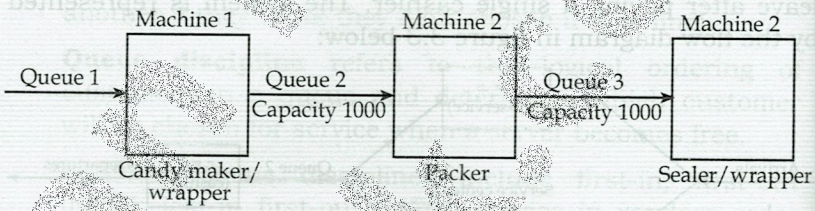

Figure 3.5: Candy-production line

Figure 3.5, the system is modeled as having three service centers, each center having $c = 1$ server (a machine), with queue-capacity constraints between machines. It is assumed that a sufficient supply of raw material is always available at the first queue. Because of the queue-capacity constraints, machine 1 shuts down whenever the inventory buffer fills to capacity, while machine 2 shuts down whenever the buffer empties. In brief, the system consists of three single-server queues in series with queue-capacity constraints and a continuous arrival stream at the first queue.

QUEUEING NOTATION

Recognizing the diversity of queueing systems, Kendall [1953] proposed a notational system for parallel server systems which has been widely adopted. An abridged version of this convention is based on the format *A /B / c / N / K*. These letters represent the following system characteristics:

A represents the inter-arrival time distribution. *B* represents the service-time distribution. [Common symbols for *A* and *B* include *M* (exponential or Markov), *D* (constant or deterministic), E_k (Erlang of order *k*), *PH* (phase-type), *H* (hyper exponential), G (arbitrary or general), and *GI* (General independent).]

c represents the number of parallel servers. N represents the system capacity. K represents the size of the calling population

For example, $M / M / 1 / \infty / \infty$ indicates a single-server system that has unlimited queue capacity and an infinite population of potential arrivals. The inter arrival times and service times are exponentially distributed. When N and K are infinite, they may be dropped from the notation. For example, M / M / 1 / ∞ / ∞ is often shortened to $M/M/1$.

Additional notation used for parallel server systems is listed in Table 3.2 given below. The meanings may vary slightly from system to system. All systems will be assumed to have a FIFO queue discipline.

Table 3.2. Queueing Notation for Parallel Server Systems

P_n	Steady-state probability of having n customers in system
$P_n,(t)$	Probability of n customers in system at time t
λ	Arrival rate
λ_e	Effective arrival rate
μ	Service rate of one server
ρ	Server utilization
A_n	Interarrival time between customers n–1 and n
S_n,	Service time of the nth arriving customer
W_n	Total time spent in system by the nth arriving customer
W_n^Q	Total time spent in the waiting line by customer n .
L(t)	The number of customers in system at time t.
L Q(t)	The number of customers in queue at time t
L	Long-run time-average number of customers in system
L_Q	Long-run time-average number of customers in queue
w	Long-run average time spent in system per customer
w_Q	Long-run average time spent in queue per customer

SIMULATION OF SINGLE SERVER AND MULTIPLE SERVER QUEUING SYSTEMS

The Single-Server Queue

A queueing system is described by its calling population, the nature of the arrivals, the service mechanism, the system capacity, and the queueing discipline. A single-channel queueing system is portrayed in figure 3.5

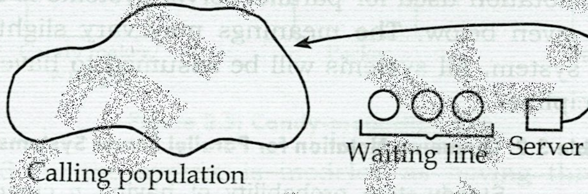

Figure 3.6: Queueing System

In the single-channel queue, the calling population is infinite; that is, if a unit leaves the calling population and joins the waiting line or enters service, there is no change in the arrival rate of other units that may need service.

Arrivals for service occur one at a time in a random fashion; once they join the waiting line, they are eventually served. In addition, service times are of some random length according to a probability distribution which does not change over time.

The system capacity; has no limit, meaning that any number of units can wait in line.

Finally, units are served in the order of their arrival by a single server or channel.

Arrivals and services are defined by the distributions of the time between arrivals and the distribution of service times, respectively.

For any simple single or multi-channel queue, the overall effective arrival rate must be less than the total service rate, or the waiting line will grow without bound. When queues grow without bound, they are termed **"explosive"** or **unstable**.

The state of the system is the number of units in the system and the status of the server, busy or idle.

An event is a set of circumstances that cause an instantaneous change in the state of the system. In a single –channel queueing system there are only two possible events to affect the state of the system.

They are the entry of a unit into the system and the completion of service on a unit. The queueing system includes the server, the unit being serviced, and units in the queue. The simulation clock is used to track simulated time. If a unit has just completed service, the simulation proceeds in the manner shown in the flow diagram of figure 3.6. Note that the server has only two possible states: it is either busy or idle.

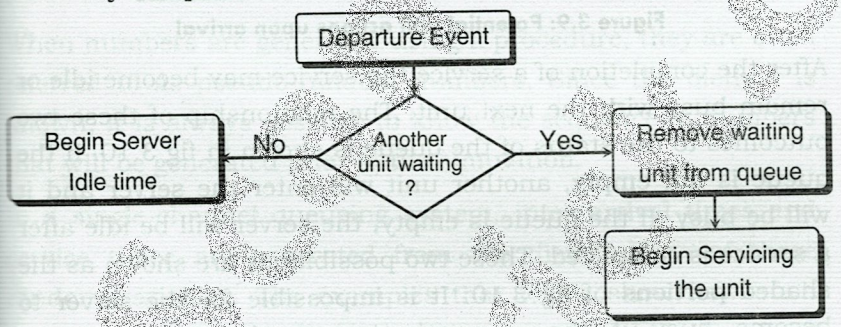

Figure 3.7: Service-just-completed flow diagram

The arrival event occurs when a unit enters the system. The flow diagram for the arrival event is shown in figure 3.7. The unit may find the server either idle or busy; therefore, either the unit begins service immediately, or it enters the queue for the server. The unit follows the course of action shown in fig 3.8.

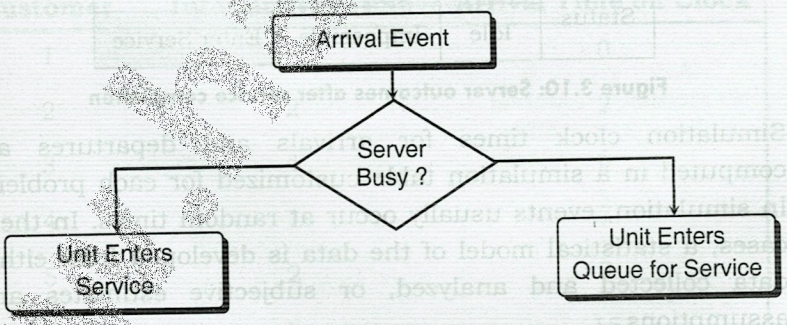

Figure 3.8: Unit-Entering system flow diagram

If the server is busy, the unit enters the queue. If the server is idle and he queue s empty, the unit begins service. It is not possible for the server to be idle and he queue to be nonempty.

		Queue Status	
		Not Empty	Empty
Server Status	Busy	Enter Queue	Enter Queue
	Idle	Impossible	Enter Service

Figure 3.9: Potential unit actions upon arrival

After the completion of a service the service may become idle or remain busy with the next unit. The relationship of these two outcomes to the status of the queue is shown in fig 3.10. If the queue is not empty, another unit will enter the server and it will be busy. If the queue is empty, the server will be idle after a service is completed. These two possibilities are shown as the shaded portions of fig 3.10. It is impossible for the server to become busy if the queue is empty when a service is completed. Similarly, it is impossible for the server to be idle after a service is completed when the queue is not empty.

		Queue Status	
		Not Empty	Empty
Server Status	Busy		Impossible
	Idle	Impossible	Enter Service

Figure 3.10: Server outcomes after service completion

Simulation clock times for arrivals and departures are computed in a simulation table customized for each problem. In simulation, events usually occur at random times. In these cases, a statistical model of the data is developed from either data collected and analyzed, or subjective estimates and assumptions.

Random numbers are distributed uniformly and independently on the interval l (0, 1). Random digits are uniformly distributed on the set {0, 1, 2... 9}. Random digits can be used to form random numbers by selecting the proper number of digits for each random number and placing a decimal point to the left of the value selected. The proper number of digits is dictated by the accuracy of the data being used for input purposes. If he n put distribution has values with two decimal places, two digits are taken from a random-digits table and the decimal point is placed to the left to form a random number.

When numbers are generated using a procedure, they are often referred to as pseudo-random numbers. Since the method is known, it is always possible to know the sequence of numbers that will be generated prior to the simulation.

In a single-channel queueing system, inter arrival times and service times are generated from the distributions of these random variables. The examples that follow show how such times are generated. For simplicity, assume that the times between arrivals were generated by rolling a die five times and recording the up face. Table 3.3 contains a set of five inter arrival times are used to compute the arrival times of six customers at the queuing system.

Table 3.3: Inter arrival and Clock Times

Customer	Inter arrival Time	Arrival Time on Clock
1	--	0
2	2	2
3	4	6
4	1	7
5	2	9
6	6	15

The first customer is assumed to arrive at clock time 0. This starts the clock in operation. The second customer arrives two time units later, at a click time of 2. The third customer arrives four time units later, at a clock time of 6; and so on.

The second time of interest is the service time. The only possible service times are one, two, three, and four time units. Assuming that all four values are equally likely to occur, these values could have been generated by placing the numbers one through four on chips and drawing the chips from a hat with replacement, being sure to record the numbers selected.

Now, the inter arrival times and service times must be meshed to simulate the single-channel queueing system. As shown in table 3.4, the first customer arrives at clock time 0 and immediately begins service, which requires two minutes. Service is completed at clock time 2. The second customer arrives at clock time 2 and is a finished at clock time 3. Note that the fourth customer arrived at clock time 7, but service could not begin until clock time 9. This occurred because customer 3 did not finish service until clock time 9.

Table 2 was designed specifically for a single-channel queue which serves customers on a first-in, first-out (FIFO) basis. It keeps track of the clock time at which each event occurs. The second column of table 2 records the clock time of each arrival even, while the last column records the clock time of each departure event.

Table 3.4: Simulation Table emphasizing Clock Times

A Customer No.	B Arrival Time (Clock)	C Time Service Begins (Clock)	D Service Time (Duration)	E Time Service Ends (Clock)
1	0	0	2	2
2	2	2	1	3
3	6	6	3	9
4	7	9	2	11
5	9	11	1	12
6	15	15	4	19

Example:

A small g
Customers
8 minutes
the same p
1 to 6 min
problem is
service of 2

Service Ti (Min)
1
2
3
4
5
6

A simulatio
system is
system fron
reached.

A set of un
generate th
have the fol

1. The s
betwe
2. Succe

The time-be
Note that
correspondi
of table 3.4
table. Alte
cumulative
minutes as

Example: Single-Channel Queue

A small grocery store has only one checkout counter. Customers arrive at this checkout counter at random from 1 to 8 minutes apart. Each possible value of inter-arrival time has the same probability of occurrence. The service times vary from 1 to 6 minutes with the probabilities shown in table 3.5. The problem is to analyze the system by simulating the arrival and service of 20 customers.

Table 3.5: Service Time Distribution

Service Time (Min)	Probability	Cumulative Frequency	Random Digit Assignment
1	0.10	0.10	01-10
2	0.20	0.30	11-30
3	0.30	0.60	31-60
4	0.25	0.85	61-85
5	0.10	0.95	86-95
6	0.05	1.00	96-00

A simulation of a grocery store that starts with an empty system is not realistic unless the intention is to model the system from startup or to model until steady-state operation is reached.

A set of uniformly distributed random numbers is needed to generate the arrivals at he checkout counter. Random numbers have the following properties:

1. The set of random numbers is uniformly distributed between 0 and 1.
2. Successive random numbers are independent.

The time-between-arrival determination is shown in table 3.6. Note that the first random digits are 913. To obtain the corresponding time between arrivals, enter the fourth column of table 3.4 and read 8 minutes from the first column of the table. Alternatively, we see that 0.913 is between the cumulative probabilities 0.876 and 1.000, again resulting in 8 minutes as the generated time

Table 3.6: Time between Arrivals Determination

Customers	Random Digits	Time Between Arrivals (Min)	Customers	Random Digits	Time Between Arrivals (Min)
1	–	–	11	109	1
2	913	8	12	093	1
3	727	6	13	607	5
4	015	1	14	738	6
5	948	8	15	359	3
6	309	3	16	888	8
7	922	8	17	106	1
8	753	7	18	212	2
9	235	2	19	493	4
10	302	3	20	535	5

Service times for all 20 customers are shown in table 3.7. These service times were generated based on the methodology described above, together with the aid of table 3.5. The first customer's service time is 4 minutes because the random digits 84 fall in the bracket 61-85, or alternatively because the derived random number 0.84 falls between the cumulative probabilities 0.61 and 0.85.

Table 7: Service Times Generated

Customer	Random Digits	Service Time (Min)	Customer	Random Digits	Service Time (Min)
1	84	4	11	32	3
2	10	1	12	94	5
3	74	4	13	79	4
4	53	3	14	05	1
5	17	2	15	79	5
6	79	4	16	84	4
7	91	5	17	52	3
8	67	4	18	55	3
9	89	5	19	30	2
10	38	3	20	50	3

The essence of a manual simulation is the simulation table. These tables are designed for the problem at hand, with columns added to answer the questions posed. The simulation table for the single-channel queue, shown, in table 3.8 that is an extension of Table 3.2. The first step is to initialize the table by filling in cells for the first custom r.

The first customer is assumed to arrive at time 0. Service begins immediately and finishes at time 4. The customer was in the system for 4 minutes. After the first customer, subsequent rows in the table are based on the random numbers for inter arrival time and service time and the completion time of the previous customer. For example, the second customer arrives at time 8. Thus, the server was idle for 4 minutes. Skipping down to the fourth customer, it is seen that this customer arrived at time 15 but could not be served until time 18. This customer had to wait in the queue for 3 minutes. This process continues for all 20 customers.

Table 3.8: Simulation Table for the queueing problem

Customer	Inter arrival Time (Min)	Arrival Time	Service Time (Min)	Time Service Begins	Waiting Time in Queue (Min)	Time Service Ends	Time Customer Spends in System (Min)	Idle Time of Server (Min)
1		0	4	0	0	4	4	
2	1	1	2	4	3	6	5	0
3	1	2	5	6	4	11	9	0
4	6	8	4	11	3	15	7	0
5	3	11	1	15	4	16	5	0
6	7	18	5	18	0	23	5	2
7	5	23	4	23	0	27	4	0
8	2	25	1	27	2	28	4	1
9	4	29	4	29	0	33	4	1
10	1	30	3	33	3	36	6	0
12	4	34	5	36	2	41	7	0
13	7	45	4	45	0	49	4	1
14	6	51	5	51	0	56	5	2
15	3	54	3	56	2	59	5	0
16	8	62	2	62	0	64	2	3
17	8	70	4	70	0	74	4	6
18	2	72	3	74	2	77	5	0
19	7	79	1	79	0	80	1	2
20	4	83	2	83	0	85	2	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
100	$\frac{5}{415}$	415	$\frac{2}{317}$	416	$\frac{1}{174}$	418	$\frac{3}{491}$	$\frac{0}{101}$

1. The average waiting time for a customer is 2.8minutes. This is determined in the following manner:

   Average Waiting Time

   $$= \frac{\text{Total time customers wait in queue (min)}}{\text{Total no. of customers}}$$

   $$= \frac{56}{20} = 2.8 \text{ minutes}$$

2. The probability that a customer has to wait in the queue is 0.65. This is determined in the following manner:

   $$\text{Probability (wait)} = \frac{\text{Number of customers who wait}}{\text{Total number of customers}}$$

   $$= \frac{13}{20} = 0.65$$

3. The fraction of idle time of the server is 0.21. This is determined in the following manner:

   Probability of idle server

   $$= \frac{\text{Total idle time of server (minutes)}}{\text{Total run time of simulation (minutes)}}$$

   $$= \frac{18}{86} = 0.21$$

   The probability of the server being busy is the complement of 0.21, or 0.79.

4. The average service time is 3.4 minutes, determined as follows:

   $$\text{Average service time (minutes)} = \frac{\text{Total service time}}{\text{Total number of customers}}$$

   $$= \frac{68}{20} = 3.4 \text{ minutes}$$

   This result can be compared with the expected service time by finding the mean of the service-time distribution using the equation

   $E(s) = \sum sp(s)$

Applying the expected-value equation to the distribution in table 2.7 gives an expected service time of:

= 1(0.10) + 2(0.20) + 3(0.30) + 4(0.25) + 5(0.10) + 6(0.50)

= 3.2 minutes

The expected service time is slightly lower than the average time in the simulation. The longer simulation, the closer the average will be to E (S).

5. The average time between arrivals is 4.3 minutes. This is determined in the following manner:

Average time between arrivals (minutes)

$$= \frac{\text{Sum of all times between arrivals (minutes)}}{\text{Number of arrivals - 1}}$$

$$= \frac{82}{19} = 4.3 \text{ minutes}$$

One is subtracted from the denominator because the first arrival is assumed to occur at time 0. This result can be compared to the expected time between arrivals by finding the mean of the discrete uniform distribution whose endpoints are a = 1 and b = 8. The me is given by

$$E(A) = \frac{a + b}{2} = \frac{1 + 8}{2} = 4.5 \text{ minutes}$$

The expected time between arrivals is slightly higher than the average. How v r, as the simulation becomes longer, the average value of the time between arrivals will approach the theoretical mean, E (A).

6. The average waiting time of those who wait is 4.3 minutes. This is determined in the following manner:

$$\text{Average waiting time of Those who wait (minutes)} = \frac{\text{Total time customers wait in queue}}{\text{Total number of customers who wait}}$$

$$= \frac{56}{13} = 4.3 \text{ minutes}$$

7. The average time a customer spends in the system is 6.2 minutes. This can be determined in two ways. First, the computation can be achieved by the following relationship:

$$\text{Average time customer Spends in the system} = \frac{\text{Total time customers spend in system}}{\text{Total number of customers}}$$

$$= \frac{124}{20} = 6.2 \text{ minutes}$$

The second way of computing this same result is to realize that the following relationship must hold:

$$\text{Average time customer Spends in the system} = \text{spends waiting in the queue} + \text{average time customer spends in service}$$

From findings 1 and 4 this results in:

Average time customer spends in the system

= 2.8+ 3.4 = 6.2 minutes.

EXAMPLE: The Able Baker Carhop Problem

This example illustrates the simulation procedure when there is more than one service channel. Consider a drive-in restaurant where carhops take orders and bring food to the car. Cars arrive in the manner shown in table 3.9. There are two carhops-Able and Baker. Able is better able to do the job and works a bit faster than Baker. The distribution of their service times are shown in tables 3.10 and 3.11

Table 3.9: Interarrival distribution of Cars

Time Between arrivals (Min)	Probability	Cumulative Probability	Random Digit Assignment
1	0.25	0.25	01-25
2	0.40	0.65	26-65
3	0.20	0.85	66-85
4	0.15	1.00	86-00

The simulation proceeds in a manner similar to example 1, except that it its more complex because of the two servers. A simplifying rule is that Able gets the customer if both carhops are idle. Perhaps, Able has seniority. (The solution would be different if the decision were made at random or by any other rule.)

Table 3.10: Service Distribution of Able

Service Time	Probability	Cumulative Probability	Random-Digit Assignment
2	0.30	0.30	01-30
3	0.28	0.58	31-58
4	0.25	0.83	59-83
5	0.17	1.00	84-00

Table 3.11: Service Distribution of Baker

Service Time	Probability	Cumulative Probability	Random-Digit Assignment
3	0.35	0.35	01-35
4	0.25	0.60	36-60
5	0.20	0.80	61-80
6	0.20	1.00	81-00
	1.00		

Here there are more events: a customer arrives, a customer begins service from able, a customer completes service from Able, a customer begins service from Baker, and a customer completes service from Baker. The simulation can be done as in above example.

Note: This can be a class work engagement and it can be proceed as:

After the first customer, the cells for the other customers must be based on logic and formulas. For example, the "clock time of arrival" in the row for the second customer is computed as follows:

$$D_2 = D_1 + C_2$$

The logic to compute who gets a given customer, and when that service begins, is more complex. The logic goes as follows when a customer arrives: if the customer finds able idle, the customer begins service immediately with able. If able is not idle but baker is, then the customer begins service immediately with baker. If both are busy, the customer begins service with the first server to become free.

MEASUREMENT OF QUEUING SYSTEM PERFORMANCE

The primary long-run measures of performance of queueing systems are the long-run-average number of customers in the system (L) and in the queue (L_Q), the long-run average time spent in system (w) and in the queue (w_Q) per customer utilization, or proportion of time that a server in busy (ρ). the term "system" usually refers to the waiting line plus the service mechanism, but in general, can refer to any subsystem of the queueing of the queueing system; whereas the term "queue" refers to the waiting line alone. Other measures of performance of interest include the long-run proportion of customers who are delayed in queue longer than t_0 time units, the long-run proportion of customers turned away because of capacity constraints, and the long-run proportion of time the waiting line contains more than k_0 customers.

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

sample average, and a time-integrated (or time-weighted) sample average.

TIME-AVERAGE NUMBER IN SYSTEM L

Consider a queuing system over a period of time T, and L(t) denote the number of customers in the system at time t. A simulation of such a system is shown in figure 3.10.

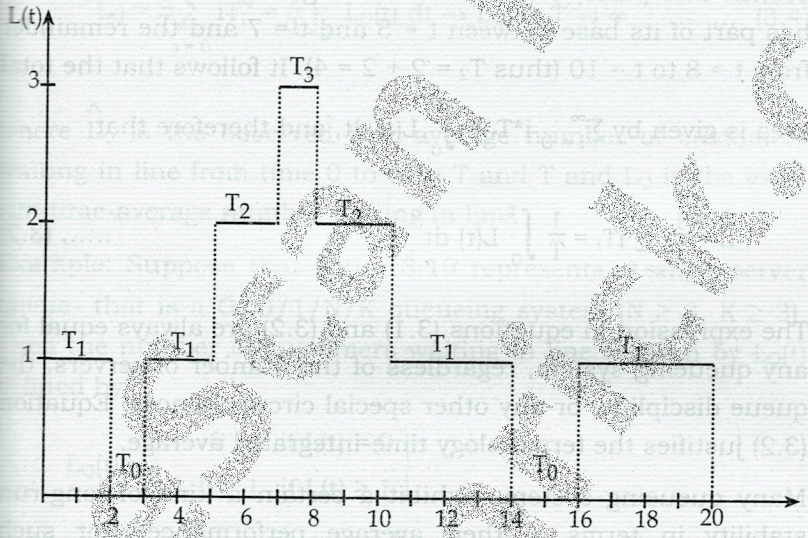

Fig: 3.10: Number in system, L(t), at time t

Let T_1 denote the total time during [0, T] in which the system contained exactly i customers. In figure 6.6, it is seen that $T_0 = 3$, $T_1 = 12$, $T_1 = 4$ and $T_3 = 1$. (The line segments whose lengths total $T_1 = 12$ are labeled "T_1" in figure etc.) In general, $\sum_{i=0}^{\infty} T_i = T$. The time-weighted-average number in a system is defined by

$$\hat{L} = \frac{1}{T} \sum_{i=0}^{\infty} iT_i = \sum_{i=0}^{\infty} i \left(\frac{T_i}{T}\right) \qquad \ldots\ldots (3.1)$$

For figure 3.10, $\hat{L} = \frac{[0(3) + 1(12) + 2(4) + 3(1)]}{20} = \frac{23}{20} = 1.15$ customers. Notice that $\frac{T_i}{T}$ is the proportion of time the system

contains exactly i customers. The estimator $\hat{L}$ is an example of a time-weighted average. By considering figure 3.10, it can be seen that the total area under the function L(t) can be decomposed into rectangles of height I and length T_i. For example, the rectangle of area $3 \times T_3$ has base running from t = 7 to t = 8 (thus $T_3 = 1$); however, most of the rectangles are broken into parts, such as the rectangle of area $2 \times T_2$ which has part of its base between t = 5 and t = 7 and the remainder from t = 8 to t = 10 (thus $T_2 = 2 + 2 = 4$). It follows that the total area is given by $\sum_{i=0}^{\infty} i*T_i = \int_0^T L(t)\,dt$, and therefore that

$$\hat{L} = \frac{1}{T}\sum_{i=0}^{\infty} iT_i = \frac{1}{T}\int_0^T L(t)\,dt \qquad \ldots\ldots (3.2)$$

The expression in equations (3.1) and (3.2) are always equal for any queueing system, regardless of the number of servers. the queue discipline, or any other special circumstances. Equation (3.2) justifies the terminology time-integrated average.

Many queueing systems exhibit a certain a kind of long-run stability in terms of their average performance. For such systems, as time T gets large, the observed time-average number in the system $\hat{L}$ approaches a limiting value, say L, which is called the long-run time-average number in system- that is with probability 1,

$$\hat{L} = \frac{1}{T}\sum_{i=0}^{\infty} L(t)dt \to L \text{ as } T \to \infty \qquad \ldots\ldots\ldots (3.3)$$

The estimator $\hat{L}$ is said to be strongly consistent for L. If simulation run length T is sufficiently long, the estimator $\hat{L}$ becomes arbitrarily close to L. Unfortunately, for $T < \infty$, $\hat{L}$ depends on the initial conditions at time 0.

Equation (3.2) and (3.3) can be applied to any subsystem of queuing system as well as they can to the whole system. If $L_Q(t)$ denotes the number of customers waiting in line, and T_i^Q denotes the total time during [0, T] in which exactly i customers are waiting in line, then

$$\hat{L}_Q = \frac{1}{T}\sum_{i=0}^{\infty} iT_i^Q = \frac{1}{T}\int_0^T L_Q(t)\,dt \to L_Q \text{ as } T \to \infty \quad \ldots\ldots\ldots (3.4)$$

where $\hat{L}_Q$ is the observed time-average number of customers waiting in line from time 0 to time T and T and L_Q is the long-run time-average number waiting in line.

Example: Suppose that figure 3.10 represents a single-server queue- that is a G/G/1/N/K queueing system ($N \geq 3$, $K \geq 3$). Then the number of customers waiting in line is given by $L_Q(t)$ defined by

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

and show in figure 3.7. thus, $T_0^Q = 5 + 10 = 15$, $T_1^Q = 2 + 2 = 4$, and $T_2^Q = 1$. Therefore,

$$\hat{L}_Q = \frac{0(15) + 1(4) + 2(1)}{20}$$

$$= 0.3 \text{ customers}$$

AVERAGE TIME SPENT IN SYSTEM PER CUSTOMER w

If we simulate a queueing system for some period of time, say T, then we can record the time each customer spends in the system during [0, T], say $W_1, W_2, \ldots, W_N$ where N is the number of arrivals during [0, T]. The average time spend in system per customer, called the average system time, is given by the ordinary sample average.

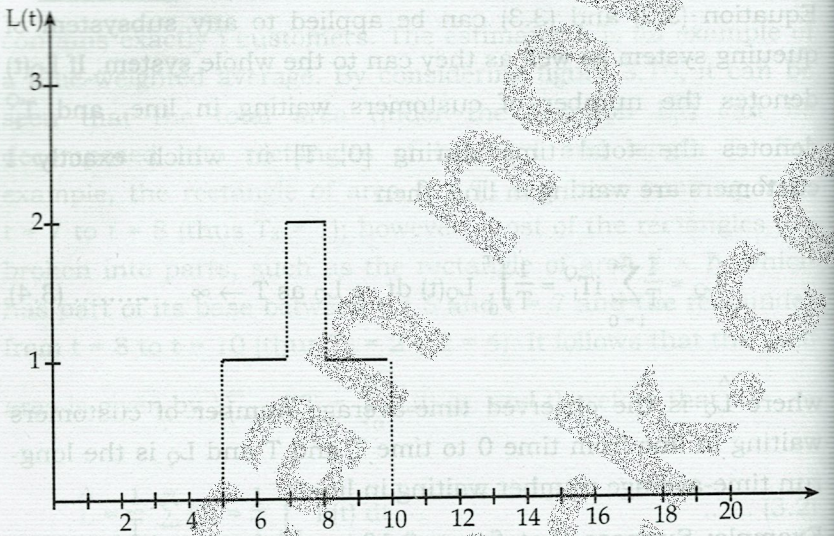

Fig: 3.11: Number waiting in line, $L_Q(t)$ at time t

$$\hat{w} = \frac{1}{N}\sum_{i=1}^{N} W_i \qquad \text{..... (3.5)}$$

For stable system, as $N \to \infty$,

$$\hat{w} \to w \qquad \text{..... (3.6)}$$

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

If the system under consideration is the queue alone, Equation (3.5) and (3.6) are written as

$$\hat{w}_Q = \frac{1}{N}\sum_{i=1}^{N} W_i^Q \to w_Q \quad \text{as} \quad N \to \infty \qquad \text{..... (3.7)}$$

where W_i^Q is the total line customer i spends waiting in queue $\hat{w}$ is the observed average time spent in queue (called delay), and w_Q is the long-run average delay per customer. The estimator $\hat{w}$ and $\hat{w}_Q$ are influenced by initial conditions at time 0 and the run length T, analogously to $\hat{L}$.

Example: For the system history shown in figure 3.10, N = 5 customers arrive, $W_1 = 2$ and $W_5 = 20 - 16 = 4$, but W_2, W_3 and W_4 cannot be computed unless more is known about the system. Assume that the system has a single server and a FIFO queue discipline. This implies that customers will depart from the system in the same order in which they arrived. Each jump upward of L(t) in figure 3.10 represents an arrival. Arrival occur at time 0, 3 , 5, 7, and 16. Similarly, departure occur at times 2, 8, 10 and 14. (A departure may or may not have occurred at time 20). Under these assumptions, it is apparent that $W_2 = 8 - 3 = 5$, $W_3 = 10 - 5 = 5$, $W_4 = 14 - 7 = 7$, and therefore

$$\hat{w} = \frac{2 + 5 + 5 + 7 + 4}{5} = \frac{23}{5} = 4.6 \text{ time units}$$

Thus, on the average, an arbitrary customer spends 4.6 time units in the system. As for time spent in the waiting line, it can be computed that $W_2^Q = 0$, $W_2^Q = 0$, $W_3^Q = 8 - 5 = 3$, $W_4^Q = 10 - 7 = 3$, and $W_5^Q = 0$; thus,

$$\hat{w}_Q = \frac{0 + 0 + 3 + 3 + 0}{5} = 1.2 \text{ time units}$$

THE CONSERVATION EQUATION: L = λw

For the system exhibited in figure 3.6, three were N = 5 arrivals in T = 20 time units, and thus the observed arrival rate was $\hat{\lambda} = \frac{N}{T} = \frac{1}{4}$ customer per time unit. Recall that $\hat{L} = 1.15$ and $\hat{w} = 4.6$; hence it follows that

$$\hat{L} = \hat{\lambda}\hat{w} \qquad \ldots\ldots (3.8)$$

This relationship between L, λ, and w is not coincidental; it holds for almost all queueing systems or subsystems regardless of the number of servers, the queue discipline, or any other special circumstances. Allowing $T \to \infty$ and $N \to \infty$, equation 3.8 becomes

$$L = \lambda w \qquad \ldots\ldots\ldots (3.9)$$

where $\hat{\lambda} \rightarrow \lambda$, and 1 is the long-run average arrival rate. Equation (3.9) is called an conservation equation and is usually attributed to Little [1961]. It says that the average number of customers in the system of customers in the system at an arbitrary point in time is equal to the average number of arrivals per time unit, times the average time spent in the system. For figure 3.6, there is one arrival every 4 times units (on the average) and each arrival spends 3.6 time units in the system (on the average), so at an arbitrary point in time there will be $\left(\frac{1}{4}\right)$ (4.6) = 1.15 customers present (on the average).

Equation (3.8) can also be arrived by reconsidering figure 3.6 in the following manner: figure 3.8 shows system history, L(t), exactly as in the figure 3.6 with each customer's time in the system W_i, represented by a rectangle. This representation again assumes a single-server system with a FIFO queue discipline. The rectangle for the third and fourth customers are in two and three separate pieces, respectively. the i^{th} rectangle has height 1 and length W_i for each i = 1, 2, ..., N. It follows that the total system time of all customers is given by the total area under the number-in-system function, L(t); that is,

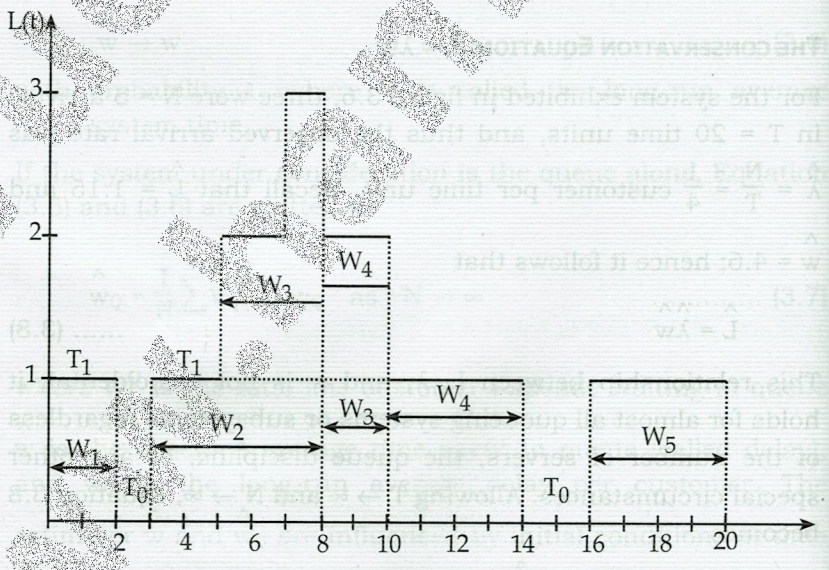

Fig: 3.12: System times, W_i, for single-server FIFO system

$$\sum_{i=1}^{N} W_i = \int_0^T L(t)\,dt \qquad \ldots\ldots\ldots (3.10)$$

Therefore, by combining Equation (3.2) and (3.5) with $\hat{\lambda} = \frac{N}{T}$, it follows that

$$\hat{L} = \frac{1}{T}\int_0^T L()\,dt$$

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

$$= \hat{\lambda}\hat{w}$$

SERVER UTILIZATION

Server utilization is defined as the proportion of time that a server is busy. Observed server utilization, denoted by $\hat{\rho}$ is defined over a specified time interval [0, T]. Long-run server utilization is denoted by ρ. For systems that exhibit long-run stability.

$$\hat{\rho} \to \rho \text{ as } T \to \infty$$

Example: Per figure 3.6 or 3.8 and assuming that the system has a single server, it can be seen that the server utilization is

$$\hat{\rho} = \frac{\text{(total buys time)}}{T} = \frac{\left(\sum_{i=i}^{\infty} T_i\right)}{T} = \frac{17}{20}.$$

MULTI-SERVER QUEUING SYSTEM

Suppose that there are c channels operating in parallel. Each of these channels has an independent and identical service-time distribution, with mean $\frac{1}{\mu}$. The arrival process is Poisson with rate λ. Arrivals will join a single queue and enter the first available service channel. The queueing system is shown in the

figure 3.13. If the number in system is n < c, an arrival will enter an available channel. However, when n ≥ c, a queue will build if arrivals occur.

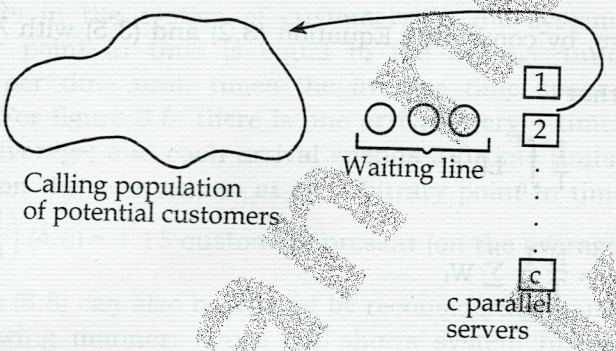

Fig: 3.13: Multi-server queueing system

The offered load is defined by $\frac{\lambda}{\mu}$. If $\lambda \geq c\mu$, the arrival rate is greater than or equal to the maximum service rate of the system (the service rate when all servers are busy); thus, the system cannot handle the load put upon it, and therefore it has no statistical equilibrium. If $\lambda > c\mu$, the waiting line grows in length at the unit but are leaving the system at a maximum rate of $c\mu$ per time unit.

For the M/M/c queue to have statistical equilibrium, the offered load must satisfy $\frac{\lambda}{\mu} < c$, in which case $\frac{\lambda}{(c\mu)} = \rho$, the server utilization. Most of the measures of performance can be expressed fairly simply in terms of P_0, the probability that the system in empty, or $\sum_{n=c} P_n$, the probability that all servers are busy, denoted by $P(L(\infty) \geq c)$, where $L(\infty)$ is a random variable representing the number in system in statistical equilibrium (after a very long time). Thus, $P(L(\infty) = n) = P_n$, for P_0 is somewhat more complex than in the previous cases. However, P_0 depends only on c and ρ.

Notice that the
number of cus
expression L –

NETWORK OF QUE

We have emp
G/G/c/N/K ty
modeled as ne
departing from
following result
population and

1. Provided t
queue, the
as the arr

2. If custom
fraction 0
departure
$\lambda_i p_{ij}$ over t

3. The overa
arrival ra
outside th

$\lambda_j = a_j + \sum_{\text{all}}$

4. If queue j
μ_j, then th

$\rho_j = \frac{\lambda_j}{c_j \mu_j}$

and $\rho_j < 1$

Notice that the average number of busy servers, or the average number of customers being served, is given by the simple expression $L - L_Q = \frac{\lambda}{\mu} = c\rho$.

NETWORK OF QUEUES

We have emphasized the study of single queues of the G/G/c/N/K type. However, many systems are naturally modeled as networks of single queues in which customers departing from one queue may be routed to another. The following results assume a stable system with infinite calling population and no limit on system capacity:

1. Provided that no customers are crated or destroyed in the queue, then the departure rate out of a queue is the same as the arrival rate into the queue, over the long run.

2. If customers, then the arrive to queue i at rate λ_I and a fraction $0 \le p_{ij} \le 1$ of them are routed to queue j upon departure, then the arrival rate from queue i to queue j is $\lambda_i p_{ij}$ over the long run.

3. The overall arrival rate into queue j, λ_j is the sum of the arrival rate from all sources. If customers arrive from outside the network at rate a_j, then

$$\lambda_j = a_j + \sum_{\text{all } i} \lambda_i p_{ij}$$

4. If queue j has $c_j < \infty$ parallel servers, each working at rate μ_j, then the long-run utilization of each server is

$$\rho_j = \frac{\lambda_j}{c_j \mu_j}$$

and $\rho_j < 1$ is required for the queue to be stable.

5. If for each queue j arrivals from outside the network form a Poisson process with rate a_j, and if there are c_j identical servicers delivering exponentially distributed service times with mean $\frac{1}{\mu_j}$ (where c_j may be ∞), then in steady state, queue j behaves like and $\frac{M}{M}{c_j}$ queue with arrival rate

$\lambda_j = a_j + \sum_{\text{all } i} \lambda_i p_{ij}$.

APPLICATIONS OF QUEUING SYSTEM

Queuing system are used in our daily life in every aspect. Some of the common applications are:

1. Commercial Queuing Systems
   - Commercial organizations serving external customers
   - Eg: Dentist, Bank, ATM, Gas Station, Plumber, Garage
2. Transportation Service Systems
   - Vehicles are customers or servers
   - Eg: Vehicles waiting at toll stations and traffic lights, trucks or ships waiting to be loaded, taxi cabs, fire engines, buses etc.
3. Business- internal service systems
   - Customers receiving service are internal to the organization providing the service
   - Eg: Inspection stations, conveyor belts, customer support etc.
4. Social Service systems
   - Eg: Judicial process, hospital, waiting list for organ transplants or students dorm rooms etc.

DISCUSSION EXERCISE

1. Explain the queuing system in simulation.
2. Explain the following queuing system characteristics:
   (a) Calling population
   (b) System capacity
   (c) Arrival process
   (d) Queue behavior and discipline
   (e) Service time and service mechanism
3. Describe Kendal-Lee notation for a queuing system
4. Explain the Inventory System in simulation.
5. Explain with suitable examples : (a) Inter-arrival time (b) Service time (c) Utility Time (d)Idle time of a queuing system
6. With a suitable flow chart describe two server queue system
   (a) A problem on News Paper Sellers.
   (b) A problem on Simulation of a (M,N) inventory system.
   (c) A problem on Single-Channel Queue.
   (d) A problem on Able Bakers carhop.
7. Explain the concept of Discrete-Event Simulation.
8. Explain in detail the event scheduling/time advance algorithm
9. Prepare a simulation table for a single channel queue system until the clock reaches time 20.

   The stopping event will be at time

   Inter-arrival times 4 5 2 8 3 6

   Service times 3 5 4 6 1 5

10. Provide the detailed flow chart of a typical arrival event and a departure event in a single channel queuing system

11. Describe Kendal notation for a queuing system.

12. Define congestion in a queuing system and describe its major characteristics.

13. What is simulation clock? Explain different time advancement mechanism with diagram.

14. What do you understand by queuing system? Describe briefly the characteristics of queuing system with the concept of queuing behaviour and queuing discipline.

15. Define single channel queuing system. What are Kendal notations used in queuing system.

16. What do you mean by Queuing system? What are the long run performance measures in a queuing model? Calculate long run time-average number of customer in the system and long run time-average number of customers in queue with the help of arbitrary example.

❑❑❑

CHAPTER OUT

After studying thi

- Features
- Process Exampl
- Applications