Introduction

CHAPTER –I

INTRODUCTION

Queueing theory is a branch of applied mathematics utilizing concepts from the field of stochastic processes. The subject grew out of the pioneering work done in this field by A.K. Erlang (1878-1929), a Danish telephone Engineer. The theory has been developed to predict fluctuating demands and enable to provide adequate service to the customers with tolerable waiting. For detailed study of classical queueing models, one can refer the contributions made by Saaty(1961), Gross & Harris (1985) and Medhi (2006).

A classical queueing system may be described as one having a service facility at which units of some kind (generally called 'customers') arrive for service and, whenever there are more units in the system than the service facility can handle simultaneously, a queue or waiting line, develops. The waiting units take their turn for service according to a pre assigned rule and after service they leave the system.

A classical queueing system is characterized by :

Arrival pattern of customers:

It describes the manner in which the arrivals occur and it is specified by the inter arrival time between any two consecutive arrivals. The input pattern also indicates whether the arrivals occur singly or in batches, if in batches, the manner in which these batches are constituted.

Service pattern:

It describes the manner in which the service is rendered and it specifies the time taken to complete a service. Some times service may be rendered in

batches, in such case, the manner of formation of batches for service has to be specified.

The number of servers:

A system may have a single server or a number of parallel servers. An arrival who finds more than one free server may choose at random any of them for receiving service. If he finds all the servers busy, he joins a queue .

The capacity of the system:

A system may have an infinite capacity (i.e.,) the queue in front of the server(s) may grow to any length. In some queueing process there is a physical limitation to the amount of waiting room, so that when the line reaches a certain length, no further customers are allowed to enter until space becomes available . These are referred to as finite queues.

Queue discipline:

The queue discipline indicates the manner in which the units are taken for service. The usual queue discipline is first come first served or FCFS. When arrivals occur in batches and service is offered individually, then the manner, in which customers arriving in batches are arranged for service, is also to be indicated.

1.1 Some special types of queueing models:

1.1.1 Bulk (or) Batch arrival queueing models:

The batch arrival is described as the flow of arrivals in batches. Gaver (1959) introduced bulk arrival queues, where the arrivals could be in batch. The literature on bulk queues with bulk arrival and with bulk service is now quite vast. Choudhury and Templeton (1983) and Medhi (1984) discuss this subject at great length.

Markovian Bulk arrival system:

In bulk input queueing models, it is assumed that the arrival streams form a Poisson process and the actual number of customers in any arriving module is a random variable which may take on any positive integral value $k<\infty$ with probability g_k. If λ_k is the arrival rate of a Poisson process of batches of size k then $g_k = (\lambda_k / \lambda)$, where $\lambda = \sum_{k=1}^{\infty} \lambda_k$ is the composite arrival rate of all batches. The total process which arises from the over lap of the set of Poisson process with rates $\{\lambda_k, k= 1,2,3 \ldots\ldots\}$ is called multiple or composite Poisson processes. The analysis of queueing models of bulk arrival $M^x/M/1$, $M^x/G/1$ and $M^x/M/\infty$ is found in Gross and Harris (1985), Medhi (1981) etc.,

1.1.2 N policy queueing models:

In classical single server queueing models an idle server will start his service as soon as a customer enters the empty system. In many situations it is important to decide when the server should start his service as, frequent setups inevitably make the operating cost too high. Among many control policies the N-policy is the most general one. In N-policy the server does not start his service until there are N – customers in the queue. This policy is introduced by Yadin and Naor (1963) and is designed to minimize server switch overs and to avoid excessive frequent use of setups.

The N-policy queueing model is used in Wireless Sensor Network (WSN) applications in which the packets are transmitted from the sensor node to base station only if the node's buffer is filled at least with 'N' packets called queue threshold. Here, based on queue threshold (N), the energy consumed by the sensor node varies and the minimum amount of energy is consumed for optimal threshold (N^*). Thus, by using an optimal queue threshold, the network lifetime can be increased to longer extent.

1.1.3 Queues with servers vacations:

Queueing systems with server classical vacations are characterized by the fact that the idle time of the server may be used for other secondary jobs. The vacation process in queueing models is governed by a vacation policy which is characterized by (1) vacation startup rule (2) vacation terminating rule and (3) vacation duration distribution. Allowing servers to take vacation make queueing models more realistic and flexible in studying real world queueing situations. Applications arise naturally in call centers with multi task employees, telecommunications and computer networks, maintenance activities, production and quality control problems etc.,

A wide class of policies for governing the vacation mechanism has been discussed in literature. The vacation policies considered in this thesis are the following:

(a) Multiple and Single Vacation Policy

In N-policy queueing models, with server vacation, as soon as the system empties, the server leaves the system for a vacation of random length. When the server returns from the vacation and finds N or more customers, he immediately starts his service. Otherwise he takes repeated number of vacations until he finds N or more customers. This policy is called a **multiple or repeated vacation policy.**

In the case of **single vacation policy**, if the server returns from vacation and finds fewer number of customers than N, he joins the system and waits until the system size reaches or exceeds N and then begins to serve exhaustively.

When vacations are interpreted as preventive maintenance performed on the server (not breakdowns) rather than production work done elsewhere, server takes only one vacation at a time. Multiple vacation policy is adopted when the server wants to utilize his idle time for other supplementary jobs.

(b) Bernoulli Vacation Policy

In both single and multiple vacation policies considered above servers take vacations only when the system becomes empty. But in some situations, especially when the service is done in two or more phases, the maintenance of the system may be required at the completion of each service and in such cases, the service may be stopped for maintenance and overhauling, or continued, if there is no fault in the system. The overhauling may be utilized as a vacation time. Recently several authors considered the vacation policy called **Bernoulli schedule vacation,** which is characterized by the feature that, at the completion of each service, the server may take a vacation with probability p or may continue to serve next unit if any, with probability (1-p).

(c) Working vacation Policy

In working vacation queues, the server works at a lower service rate rather than completely stopping service during the vacation period. At the vacation termination epochs, if there are customers in the system, the server will start a new regular busy period. Otherwise, he takes another working vacation or joins the system and stays idle according as, he follows multiple or single working vacation policy.

There is quite a difference between working vacation queue and classical vacation queue. During a vacation, customers in the former may finish service and depart the system; however, customers in the latter can impossibly depart the system. The number of customers in the former case increase or decrease, however, the number of customers in the latter case only increase. Therefore, the working vacation models have more complicated modalities and the analysis of this kind of models is more difficult than classical vacation queues.

1.1.4 Queues with server Breakdowns

Most of the classical queueing systems assume a reliable machine or server, however in practice; the server may fail and can be repaired. This phenomena of server breakdowns, can be encountered in the area of computers, communication networks, flexible manufacturing systems etc., The performance of the system may be affected heavily by these breakdowns and limited repair capacity. Queueing systems with such unreliable stations are the topics of worth investigating from the performance prediction point of view.

1.1.5 Queues with server setup or startup time

In some of the practical situations server often requires startup operations before starting each busy period. The server startup time corresponds to the preparatory work of the server before starting the service. For example when a typical machine is to be turned on, a proper setup operation before the normal use may extend its function and the number of unsteady conditions of the machine tool may be reduced. Thus an additional amount of time of random length called SET, in order to set the system into operation mode before actual service begins is called **setup period.**

1.1.6 Two phases of service

One of the important characteristics of a queueing system is the service process. A class of systems where the service discipline involves more than one service has been receiving lot of attention recently. Various scenarios have been considered in the literature and some of them are listed below:

- Queueing system with Second optional service (SOS): The server provides First essential service (FES) to all arriving customers and as soon as the FES of a customer completes, with probability r, the customer may opt for the second service or else with probability (1-r) the customer

may leave the system and the next customer at the head of the queue (if any) is taken up for FES.

- The server provides two phases of heterogenous service one after the other to all the arriving customers. This type is a special case of (SOS) with r = 1.
- The server provides c- types of heterogeneous service and customers may choose one of the services with some probability.
- A customer chooses one type of heterogenous service in the first phase and then he takes the option to repeat or leave the system.

1.1.7 (m, N) policy or Bi-level control policy

Consider a single server queueing system in which the sever is deactivated as soon as the system empties. The server is reactivated and starts a setup when m customers accumulate in the queue. After the setup if there are less than **N (≥m)** customers in the queue then the server remains dormant in the system until the number of customers reaches N. If N or more customers are in the queue then, after the setup, the server begins to serve the customer immediately. This policy is called (m,N) policy. This (m,N) policy is more general than the usual (N,N) policy in which the server starts the setup when N customers have piled up in the queue (thus in the (N,N) policy system, the server starts his service as soon as the setup is finished). The (m,N) policy is justified when the customer waiting cost is more expensive than the server idle cost. The double-threshold policy fits into the real manufacturing settings because the setup operation does not start immediately after the machine ceases its production operation. Also if the setup cost is very high the operator may not need to wait until the accumulated items reach the usual single threshold **N**.