Queuing models: Introduction

- Queuing models are also called waiting line models.
- **Queuing** theory is the mathematical study of queuing, or waiting lines. A basic queuing system consists of an arrival process (how customers arrive at the queue), the queue itself, the service process for attending to those customers, and departures from the system
- We are quite familiar with queues in our day-to-day life. Common examples of queuing models that we encounter are going to a doctor or going to a barber shop.
- A queuing system essentially happens when there are entities or people who are called arrivals who require a kind of service from another entity. There is a service and there is a line or a queue where a person is joining this system.
- If we take a typical example of a doctor; people arrive, spend some time, get served and people leave.
- Queuing models can be of several types. First category is called a single server queuing model where there is only one server. We also have multiple server queuing models where there are multiple servers for the same job.
- A good example of a multiple server queuing model is a railway reservation system where we could have several counters where people who come join this line and whichever server is free the first person will go, then the next and then the next person goes and so on. So, we have a multiple server model where there is more than one server. Here, there is a common line and as soon as a server is free, this person will join, get the service and leave.

- Within the single server and multiple servers there are two categories. One is called a **finite queue length** and **infinite queue length**.
- The **infinite queue length** model assumes, that every person who comes will joins the line, for example, if already three people are waiting for the doctor, the fourth person will join the line and so on. There is **no restriction** on the number of people who are actually waiting or there is no restriction on the length of the queue. The queue length can theoretically be infinite so it can go on and on.
- In **finite queue length** models we try to restrict the queue length to a certain limit after which we say that if this threshold limit is reached, people who come into the system do not join the system.
- A good example of a finite queue length model is a garage. Let us say there is a garage with a single server or a single mechanic. There is a mechanic here and let us say this garage has space to park, say ten cars. For example, someone is coming into the garage to give his or her car for service if some slots are available.
- There is one further classification which is called **finite population models** and **infinite population models**. The **population** of potential customers may be assumed to be **finite** or **infinite**.
- **Finite population model**: if arrival rate depends on the number of customers being served and waiting.
- If we take the example of the doctor or the reservation system or the car mechanic they all come under the category of what are called infinite population models.
- The queuing system is also characterized by the distribution of this arrival and distribution of this service. The arrivals and services can follow any

given distribution or we can go observe physically, what is happening in the queuing system. From the data that we can collect from what is actually happening we can **fit a corresponding distribution**. Most of the times it also observed that arrivals follow a Poisson distribution, with arrival rate called lambda (λ: the arrival rate) per hour. Lambda (λ) usually denotes the arrival rate in a queuing system.

- It is also observed from practice, that service times are exponentially distributed, at the rate of mu (μ) per hour.
- In our study we will assume that arrivals follow a Poisson distribution with lambda per hour and service times are exponential, denoted by mu per hour.
- **Traffic intensity** is a measure of how busy a system is, and is defined as the u = lambda/mu.
- Now let us discuss about the relationship between this lambda and mu. There are some cases which are discussed as follows.
- Case 1: When lambda by mu is less than 1. For example, we assume lambda equal to 5 per hour and mu is equal to 6 per hour, then lambda by mu is less than 1. What happens, here lambda is equal to 5 per hour means on an average five people enter the system every hour, which means, on an average every twelve minutes a person enters the system and on an average every ten minutes a person gets served and leaves the system. So queue length will decrease as the time passes.
- Case 2: When lambda by mu is greater than one. For example, if lambda is 6 per hour and mu is 4 per hour. Then, every hour six people on an average enter the system and four people on an average leave the system, so, the queue length will increase by 2 every hour. So queue length will increase as the time passes.

- Case 3: When lambda by mu is equal to one. For example, if lambda is 6 per hour and mu is 6 per hour. Then, every hour six people on an average enter the system and six people on an average leave the system, so, the queue length will remain same as in the beginning. So queue length will remain constant as the time passes.
- Another parameter that we need to discuss is called the **queue discipline** or the **service principle**. *Queuing Discipline* represents the way the queue is organised (rules of inserting and removing customers to/from the queue). There are these ways.
- If we take the example of a doctor that we have seen let us say there are six people already in the line, you join as the seventh person So ordinarily you would know and you would expect the queue discipline to be what is called **first-in-first out**, also called first come first served, which is either **FIFO or FCFS**. Ordinarily, when human beings are involved and the service provider is also a human being it is customary to assume a first-in-first out service discipline or first-in-first out rule to send the next person into the system. (FIFO (First In First Out) also called FCFS (First Come First Serve) - orderly queue)
- There are times we can have **a last-in-first out system** and so on. (LIFO (Last In First Out) also called LCFS (Last Come First Serve) - stack)
- **SIRO** stands for service in random order. Under this type of queue structure, the customer is chosen for **service randomly** and hence all the customers are equally likely to be selected. Therefore, the time of arrival of the customer has no consequence on the selection of the customer.

  Priority Queue that may be viewed as a number of queues for various priorities.