# Lectures 19 and 20. Discrete-time queuing systems. Bernoulli single-server queuing process. Limited and unlimited capacity.

**YULIA R. GEL**

**CS/SE/STAT 3341 Probability and Statistics in Computer Science and Software Engineering**

April 4 and 6, 2014

# Motivation

Queuing theory deals with problems which involve queuing (or waiting). Typical examples might be:

- banks and supermarkets - waiting for service
- computers - waiting for a response
- failure situations - waiting for a failure to occur, e.g. in a piece of equipment
- public transport - waiting for a train or a bus

# Queuing System – More Formally

**Definition.** A **queuing system** is a facility consisting of one or several servers designed to perform certain tasks or process certain jobs and a queue of jobs waiting to be processed.

As we know queues are a common every-day experience. Queues form because resources are limited. In fact, it makes economic sense to have queues. For example,

- How many airport gates are needed to avoid long queuing of aircrafts?
- How many buses or trains would be needed if queues were to be avoided/eliminated?

# Queuing System – How Does It Work?

In designing queueing systems we aim to balance between

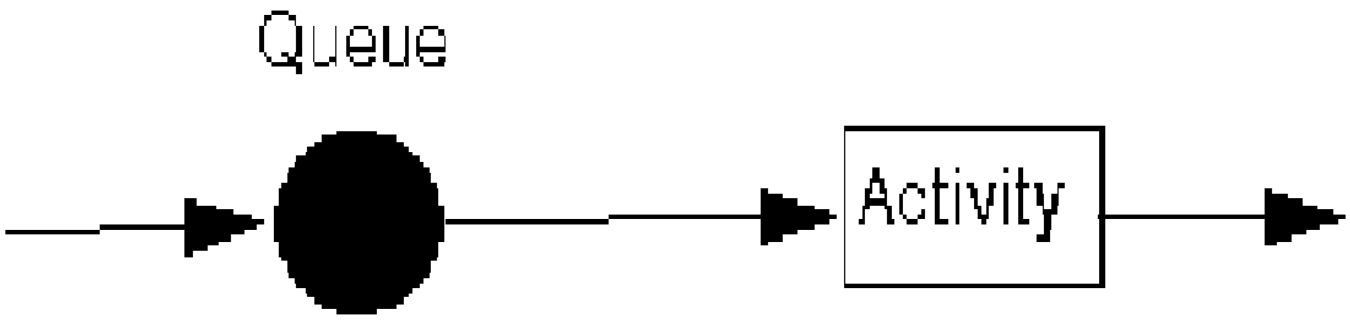
- service to customers (short queues implying many servers)

  **vs.**

- economic considerations (not too many servers).

# Queuing System – How Does It Work?

In essence all queuing systems can be broken down into individual sub-systems consisting of **entities** queuing for some **activity**:



# Queuing System – How Does It Work?

Becker and Parker (2011).

# Queuing System – Mechanisms

Typically we can talk of this individual sub-system as dealing with **customers** queuing for **service**. To analyse this sub-system we need information relating to:

- arrival process;
- service mechanism;
- queue characteristics.

# Queuing System – Arrival Process

Characterizing arrival process:

- how customers arrive, e.g. single or in groups (batch/bulk arrivals)
- how the arrivals are distributed in time, e.g. what is the probability distribution of time between successive arrivals (the interarrival time distribution)
- whether there is a finite population of customers or (effectively) an infinite number

The simplest arrival process is one where we have completely regular arrivals (i.e. the same constant time interval between successive arrivals). In turn, a Poisson stream of arrivals corresponds to arrivals at random. In a Poisson stream, successive customers arrive after intervals which are independent and exponentially distributed.

# Queuing System – Arrival Process

Certainly, it is more realistic to consider the case when jobs arrive to the queuing system at random times.

A counting process $A(t)$ tells the number of arrivals that occurred by the time $t$. In stationary queuing systems (*whose distribution characteristics do not change over time*), arrivals occur at arrival rate

$$\lambda_A = \frac{EA(t)}{t}$$

for any $t > 0$, which is the expected number of arrivals per 1 unit of time.

Then, the expected time between arrivals is

$$\mu_A = \frac{1}{\lambda_A}.$$

# Queuing System – Service Mechanism

Characterizing service mechanism:

- a description of the resources needed for service to begin
- how long the service will take (the service time distribution)
- the number of servers available
- whether the servers are in series (one queue for all servers) or in parallel (each server has a separate queue)
- whether preemption is allowed (a server can stop processing a customer to deal with another "emergency" customer)

The assumption that the service times for customers are independent and do not depend upon the arrival process is common. Another common assumption about service times is that they are exponentially distributed.

# Queuing System – Service Mechanism

Suppose that we assume that once a server becomes available, it immediately starts processing the next assigned job.

In practice, service times are random because they depend on the amount of work required by each task. The average service time is $\mu_S$.

It may vary from one server to another as some computers or customer service representatives work faster than others. The **service rate** is defined as the average number of jobs processed by a server during 1 unit of time:

$$\lambda_S = \frac{1}{\mu_S}.$$

# Queuing System – Queue Characteristics

Characterizing queue:

- how, from the set of customers waiting for service, do we choose the one to be served next (e.g. FIFO (first-in first-out) - also known as FCFS (first-come first served); LIFO (last-in first-out); randomly) (this is often called the **queue discipline**)
- do we have:
  - balking (customers deciding not to join the queue if it is too long)
  - reneging (customers leave the queue if they have waited too long for service)
  - jockeying (customers switch between queues if they think they will get served faster by so doing)
  - a queue of finite capacity or (effectively) of infinite capacity

# Queuing System – Queue Characteristics

Changing the queue discipline (the rule by which we select the next customer to be served) can often reduce congestion. Often the queue discipline "choose the customer with the lowest service time" results in the smallest value for the time (on average) a customer spends queuing.

Note here that the key to queuing situations is the idea of uncertainty in, for example, interarrival times and service times.

This means that probability and statistics are needed to analyse queuing situations.

# Queuing System – Questions

In terms of the analysis of queuing situations the types of questions in which we are interested are typically concerned with measures of system performance and might include:

- How long does a customer expect to wait in the queue before they are served, and how long will they have to wait before the service is complete?
- What is the probability of a customer having to wait longer than a given time interval before they are served?
- What is the average length of the queue?
- What is the probability that the queue will exceed a certain length?
- What is the expected utilisation of the server and the expected time period during which it will be fully occupied?

# Queuing System – More Real-World Problems

These are questions that need to be answered so that management can evaluate alternatives in an attempt to control/improve the situation. Some of the problems that are often investigated in practice are:

- Is it worthwhile to invest effort in reducing the service time?
- How many servers should be employed?
- Should priorities for certain types of customers be introduced?
- Is the waiting area for customers adequate?

In order to get answers to the above questions there are two basic approaches:

- analytic methods or queuing theory (formula based);
- simulation (computer based).

# Queuing System – Finally in Math Terminology

**Queuing process** is a stochastic process $X(t)$, in which

$states = number\ of\ jobs\ in\ the\ system\ (waiting + being\ served)$.

Arrival $\Longrightarrow X(t)$ increases by 1

End of service $\Longrightarrow X(t)$ decreases by 1

# Queuing System – Math Notations

- $\lambda_A$ is arrival rate
- $\lambda_S$ is service rate
- $\mu_A = 1/\lambda_A$ is mean interarrival time
- $\mu_S = 1/\lambda_S$ is mean service time
- $r = \lambda_A/\lambda_S = \mu_S/\mu_A$ is utilization, or arrival-to-service ratio
- $X_s(t)$ number of jobs receiving service at time $t$
- $X_w(t)$ number of jobs waiting in a queue at time t
- $X(t) = X_s(t) + X_w(t)$, the total number of jobs in the system at time $t$
- $S_k$ is service time of the $k$-th job
- $W_k$ is waiting time of the $k$-th job
- $R_k = S_k + W_k$ is the response time, i.e. the total time the $k$-th job spends in the system from its arrival until the departure

# Single-server Bernoulli queuing process

It is a queuing process with:

- one server
- unlimited capacity
- arrivals according to a Binomial counting process

$$P_A = P\{arrival\ \ during\ \ any\ \ frame\}$$

- service completions according to a Binomial counting process (while there are jobs in the system);

$$P_S = P\left\{ \begin{array}{l|l} completed\ \ service & server \\ during\ \ any\ \ frame & is\ \ busy \end{array} \right\}$$

- arrivals independent of service times

# Single-server Bernoulli queuing process

**Remark.** It is not hard to notice that fastest service is 1 frame with $P_A$ and $P_S$ being constant.

Everything learned about Binomial counting processes applies to arrivals of jobs. It also applies to service completions all the time when there is at least one job in the system. We can then deduce that

- there is a Geometric($p_A$) number of frames between successive arrivals
- each service takes a Geometric($p_S$) number of frames
- service of any job takes at least one frame
- $p_A = \lambda_A \Delta$
- $p_S = \lambda_S \Delta$

# Single-server Bernoulli queuing process – Markov Chain

Bernoulli single-server queuing process is a **homogeneous Markov chain** because probabilities $p_A$ and $p_S$ never change.

As we have discussed, the number of jobs in the system increments by 1 with each arrival and decrements by 1 with each departure.

Indeed, conditions of a Binomial process guarantee that at most one arrival and at most one departure may occur during each frame.

# Single-Server Bernoulli Queuing Process – Markov Chain

Then, we can compute all transition probabilities:

$$\begin{aligned} p_{00} &= P\{no\ \ arrivals\} = 1 - p_A \\ p_{01} &= P\{new\ \ arrivals\} = p_A \end{aligned}$$

and for all $i \geq 1$

$$\begin{aligned} p_{i,i-1} &= P\{no\ \ arrivals \cap one\ \ departure\} = (1 - p_A)p_S \\ p_{i,i} &= P\{no\ \ arrivals \cap no\ \ departure\} \\ &+ P\{one\ \ arrival \cap one\ \ departure\} = (1 - p_A)(1 - p_S) + p_A p_S \\ p_{i,i+1} &= P\{one\ \ arrival \cap no\ \ departure\} = p_A(1 - p_S) \end{aligned}$$

# Single-Server Bernoulli Queuing Process – Markov Chain

The transition probability $\infty \times \infty$-matrix is three- diagonal:

$$P = \begin{pmatrix} 1-p_A & p_A & 0 & \dots \\ (1-p_A)p_S & (1-p_A)(1-p_S)+p_Ap_S & p_A(1-p_S) & \dots \\ 0 & (1-p_A)p_S & (1-p_A)(1-p_S)+p_Ap_S & \dots \\ 0 & 0 & (1-p_A)p_S & \ddots \\ \vdots & \vdots & \ddots & \ddots \end{pmatrix}$$

All the other transition probabilities equal 0 because the number of jobs cannot change by more than one during any single frame.

# Example – Car Wash Center

**Example.** Performance of a car wash center is modeled by the single-server Bernoulli queuing process with 2-minute frames.

The cars arrive every 10 minutes, on the average. The average service time is 6 minutes. Capacity is unlimited.

If there are no cars at the center at 10 am, compute the probability that one car will be washed and another car will be waiting at 10.04 am.

# Example – Car Wash Center

**Solution.** We have $\Delta = 2$ minutes,

$$\lambda_A = \frac{1}{\mu_A} = \frac{1}{10} min^{-1}$$

and

$$\lambda_S = \frac{1}{\mu_S} = \frac{1}{6} min^{-1}.$$

Thus,

$$p_A = \lambda_A \Delta = 1/5$$

and

$$p_S = \lambda_S \Delta = 1/3.$$

# Example – Car Wash Center

**Solution.** There are 2 frames between 10.00am and 10.04am.

We need the conditional probability

$$P\{X_2 = 2|X_0 = 0\} = P\{X(10.04) = 2|X(10.00) = 0\}.$$

Since the number of cars at the wash center can change by at most 1 during each frame, this probability equals

$$p_{02}^2 = p_{01}p_{12} = p_A(p_A(1 - p_S)) = \frac{1}{5} \times \frac{1}{5} \times \frac{2}{3} = \frac{2}{75}.$$

# Limited Capacity

The number of jobs in a Bernoulli single-server queuing system may potentially reach any number.

However, many systems have limited resources for storing jobs. Then, there is a maximum number of jobs $C$ that can possibly be in the system simultaneously. This number is called **capacity**.

How does limited capacity change the behavior of a queuing system?

# Limited Capacity

Until the capacity $C$ is reached, the system operates without any limitation, as if $C = \infty$. All transition probabilities remain as we calculated before.

When the system is full, i.e. $X = C$, it can accept new jobs into its queue only if some job departs. As before, the number of jobs decrements by 1 if there is a departure and no new arrival,

$$p_{C,C-1} = (1 - p_A)p_S.$$

# Limited Capacity

In all other cases, the number of jobs remains at $X = C$. If there is no departure during some frame, but a new job arrives, this job cannot enter the system, so that

$$p_{C,C} = (1 - p_A)(1 - p_S) + p_A p_S + p_A(1 - p_S) = 1 - (1 - p_A)p_S.$$

This Markov chain has states $0, 1, \ldots, C$, its transition probability matrix is finite, any state can be reached in $C$ steps.

Hence, the Markov chain is regular, i.e. all $p_{ij}^{(h)} > 0$, and its steady-state distribution is readily available.

# Example – Customer Service

**Example.** A customer service representative can work with one person at a time and have at most one other customer waiting.

Compute the steady-state distribution of the number of customers in this queuing system at any time, assuming that customers arrive according to a Bernoulli counting process with 3-minute frames and the average interarrival time of 10 minutes, and the average service takes 15 minutes.

# Example – Customer Service

**Solution.** This is a Bernoulli queuing process with limited capacity $C = 2$, i.e. one customer getting service and one customer waiting. Its parameters are

$$\lambda_A = 1/10 min^{-1}, \quad \lambda_S = 1/15 min^{-1}, \quad \Delta = 3 min.$$

We can then compute probabilities

$$p_A = \lambda_A \Delta = 0.3, \quad p_S = \lambda_S \Delta = 0.2$$

# Example – Customer Service

**Solution.** Then,

$$p_S(1-p_A) = 0.14, \quad p_A(1-p_S) = 0.24,$$

and we have the transition probability matrix

$$P = \begin{pmatrix} 0.7 & 0.3 & 0 \\ 0.14 & 0.62 & 0.24 \\ 0 & 0.14 & 0.86 \end{pmatrix}.$$

We solve the linear system $\pi P = \pi$ under the condition $\pi_0 + \pi_1 + \pi_2 = 1$.

# Example – Customer Service

**Solution.** For the 1st equation, we get $0.7\pi_0 + 0.14\pi_1 = \pi_0$, and hence $\pi_0 = 7/15\pi_1$.

For the 2nd equation, we get $0.\pi_0 + 0.62\pi_1 + 0.14\pi_2 = \pi_1$, and hence $\pi_2 = 12/7\pi_1$.

Finally, the 3rd equation takes the form $0.24\pi_1 + 0.86\pi_2 = \pi_2$.

Using the normalizing condition $\sum_{i=0}^{2} \pi_i = 1$, we get

$$\pi_0 = 49/334 = 0.1467,$$

$$\pi_1 = 105/334 = 0.3144,$$

$$\pi_2 = 180/334 = 0.5389.$$