A Day in the Life of a Queue

CPE 360

This document provides a walk-through for our final assignment, which applies the Queue data structure to understand (and improve) a real life scenario. Feel free to work in groups of two.

1 What is a Queue?

A queue is the arrival and departure of entities that follows a strictly first-in, first out (FIFO) philosophy. An example implementation of queues in C++ should be available to you from a prior assignment. Feel free to use that to base this assignment upon.

2 The problem

Picture the typical queue at your favorite burger joint. Customers arrive at the store and form a queue, where they place an order when they get their turn. Lets assume that each customer places an order that takes anywhere between 1 and 6 minutes to fulfill. The store processes these orders in the order they arrive: first-in, first-out.

Lets assume that at any given minute, customers arrive at the store with a probability $p_a(t)$, where p_a is the probability of arrival which is a function of time-of-the-day (t). Assume the following values for $p_a(t)$:

- 8:00 am to 10:00 am, $p_a = 0.3$
- 10:00 am to 11:30 am, $p_a = 0.1$
- 11:30 am to 1:30 pm, $p_a = 0.4$
- 1:30 pm to 5:30 pm, $p_a = 0.1$
- 5:30 pm to 8:00 pm, $p_a = 0.25$
- 8:00 pm to 11:00 pm, $p_a = 0.2$
- 11:00 pm to 1:00 am, $p_a = 0.1$

Beyond forming a queue, the customers are served in a FIFO manner. Each customer randomly picks an order that ranges anywhere between a minute to 6 minutes to fulfill. The store enforces a strict FIFO order processing, regardless of how long each order takes to fulfill.

2.1 Part1: How good is this current system?

Simulate the above scenario for an entire day starting at 8:00 am until store close at 1:00 am at a granularity of **1 minute**. Customers arrive with the aforesaid probability to a single queue. Each customer arrival is a call to the enqueue function. The dequeue process is assumed to be a steady stream of order processing: the chefs move onto the next order as soon as they process the current one; i.e., at every minute, we check to see if a customer is done so we can call the dequeue function.

After the simulation, provide answers for the following:

1. Average customer *wait time*. This is the mean wait time until they can place an order, i.e., time spent standing in line to get to the person behind the counter.

2. Average customer *service time*. This is the mean time to get the order in hand from the second they enter the store, i.e., wait time + service time.

3. Average *queue length*. The mean number of people in line at any given time averaged over the entire day.

4. The *best case* and *worst case* for each of the above and *at what time* did that occur: (i) wait time, (ii) service time, and (iii) queue length.

2.2 The Optimization

The store operations center realizes that during peak hours (breakfast/lunch/dinner), customers can be better served by having two people behind the counter, i.e., have two separate counters instead of one. There are two possibilities to make this happen: (i) customers form a single line, and are free to move to the next free server behind the counter when they are at the end of the queue, or (ii) form two separate queues, one each for the two counters. Customers are free to choose any queue they want to join as soon as they walk in. Customers typically pick a queue with the least number of people.

Simulate both the alternatives above by having two separate counters to serve customers at peak hours. Some helpful tips:

- For the first option, customers arrive at the store with the same probabilities and they join the single queue. When they get to the end of the queue, they pick the next free counter to go place their order.

- For the second option, a customer arrives at the store and notices two separate queues. The customer then picks the queue with lesser customers in it and joins the line.

For both these alternatives, calculate the parameters of interest: (i) average wait time, (ii) service time, and (iii) queue length. Suggest which alternative works better, and why?