CS 350: Homework #4

Due 3PM, Tuesday, March 1

Kristel Tan (ktan@bu.edu)

February 14, 2016

Question 1

Part(a)

It makes sense to model the system described in this question as an M/M/1/K queue because it has a limited buffer size of K packets. In other words, the queue has a finite size as opposed to an M/M/1 system which has an infinite queue size.

Part(b)

$$\rho = \lambda T_S = (0.05\ packets/ms)*(20\ ms) = 1$$
$$q = \frac{K}{2}\ for\ \rho = 1 \rightarrow q = \frac{3}{2} = 1.5$$
$$\Pr(S_3) = \frac{1}{3+1} = \frac{1}{4} = 0.25$$
$$\lambda' = \lambda(1-\Pr(S_3)) = (0.05\ packets/ms)*(1-0.25) = 0.0375$$
$$T_q = \frac{q}{\lambda'} = \frac{1.5}{0.0375} = 40\,ms$$
$$T_w = T_q - T_s = 40 - 20 = 20\,ms$$
$$w = \lambda' * T_w = 0.0375*20 = 0.75\ packets$$

Part(c)

$$\Pr(blip) = \Pr(S_3) = \frac{1}{K+1}\ for\ \rho = 1 \rightarrow \frac{1}{3+1} = \frac{1}{4} = 0.25$$
$$(0.25)*(2000) = 500\ blips$$
$$\Pr(plop) = P(0) = 1-\rho = 1-1 = 0\ plops$$

Part(d)

$$\lambda' = \lambda(1-\Pr(S_3)) = (0.05\ packets/ms)*(1-0.25) = 0.0375$$
$$T_q = \frac{q}{\lambda'} = \frac{1.5}{0.0375} = 40\ ms$$

Part(e)

$$\Pr(rejection) = \Pr(S_{10}) = \frac{1}{K+1} \; for \; \rho = 1 \rightarrow \frac{1}{10+1} = \frac{1}{11}$$

$$\lambda' = \lambda(1 - \Pr(S_{10})) = (0.05 \; packets/\sec) * (1 - \frac{1}{11}) \approx 0.045$$

$$q = \frac{K}{2} = \frac{10}{2} = 5$$

$$T_q = \frac{q}{\lambda'} = \frac{5}{0.045} \approx 111 \; ms$$

Part(f)

The mean delay for a queue size of $K = 10$ is much larger than a queue size of $K = 2$. This result makes sense because a packet entering a larger queue has a greater chance of having more packets in front of it. This also means that because there is more space for a packet to wait, there is less of a chance it will be dropped. However, in a smaller queue, more packets are dropped, which does not add to the mean delay. Therefore, in general, I would expect the mean delay for packets played out to increase as K also increases.

Part(g)

Increasing the size of network buffers has its advantages and disadvantages. One of its main advantages is having the ability to store more packets or requests in the buffer. This means that less packets are dropped, which results in a greater quality of service in terms of image quality, sound quality, etc. On the other hand, however, a large network buffer can also lead to prolonged delay times and service times. Hence, if the queue size is not balanced with the incoming load, the quality of service can also degrade in characteristics related to time such as frames per second, choppiness in sound, etc.

Part(h)

I would recommend a larger value of K for a "Movie on Demand" streaming application simply because it would ensure an enhanced viewing experience during playback. When streaming media data into a small queue, more packets have the chance of being dropped, which results in a choppy and intermittent playback. One might argue this and say that the overall load time for the media content in a larger K sized queue would take longer. This may be true, but at least the quality of the playback will not suffer for its duration.

On the other hand, for a teleconferencing streaming application, I would more likely recommend a smaller value of K because it requires two-way, instant interaction. A few dropped packets every now and then may degrade the quality of the audio or video, but it is better than having to wait a very long time to hear the other person's responses when communicating with them. In other words, it would be difficult to hold a conversation with someone if you had to sit in silence for a few seconds every time the larger queue loaded more data.

Question 2

Please refer to the file `hw4_q2.py` submited with this write-up to view the source code this problem.

Within the Python source code you will find that I generated four different log files for part(a) & part(b) together, part(c), part(d), and part(e) of this problem. Because the service time for an M/M/1/K system is exponentially distributed and the lambda value given does not change, parts (a) and (b) have been combined together in terms of their empirical results. Analytical results for each part, however, have been calculated respectively.

Whenever you run the `hw4_q2.py` file, these log files will be regenerated. The log files are fairly straightforward; they document monitoring events of the queue at random points in time while giving you a look at the queue as events are being serviced or rejected. At the very end, the program also displays the overall empirical and analytical results. The following are sample screenshots generated from a run of the program.

Part(a) & Part(b)

For $K = 3$, $\lambda = 50$, $T_s = 0.020$, and simulation time = 100 &
$K = 3$, $\lambda = 50$, $T_s = 0.015$, and simulation time = 100...

```
---------------------------------
     M/M/1/K Queue Analysis
---------------------------------

p =  0.0576231008994372

pr(rej) =  0.4970131421744325

Tq =  0.06221825382014645

mean Ts =  0.020306754151360785

mean Tw =  0.010988209837028293

q =  1.5647481994194539

w =  0.5494104918514146

---------------------------------
```

```
Part a

---------------------------------
M/M/1/K Queue Analytical Analysis
---------------------------------

p =  1.0

pr(rej) =  0.25

Tq =  0.04

Ts =  0.02

mean Tw =  0.02

q =  1.5

w =  0.75

--------------------------------
```

```
Part b

---------------------------------
M/M/1/K Queue Analytical Analysis
---------------------------------

p =  0.75

pr(rej) =  0.15428571428571428

Tq =  0.027162162162162162

Ts =  0.015

mean Tw =  0.012162162162162163

q =  1.1485714285714286

w =  0.5142857142857143

--------------------------------
```

Part(c)

For $K = 3$, $\lambda = 65$, $T_s = 0.015$, and simulation time $= 100$...

```
---------------------------------
     M/M/1/K Queue Analysis
---------------------------------

p =  2.7052994670004584

pr(rej) =  0.496551724137931

Tq =  0.05906638084865293

mean Ts =  0.020113869048987786

mean Tw =  0.009622998550678857

q =  1.4868433799833323

w =  0.4811499275339428

---------------------------------
```

```
---------------------------------
M/M/1/K Queue Analytical Analysis
---------------------------------

p =  0.75

pr(rej) =  0.25

Tq =  0.03062857142857143

Ts =  0.015

mean Tw =  0.01562857142857143

q =  1.1485714285714286

w =  0.5860714285714286

--------------------------------
```

Part(d)

For $K = 3$, $\lambda = 50$, T_s is fixed and $= 0.015$, and simulation time $= 100$...

```
--------------------------------
     M/D/1/K Queue Analysis
--------------------------------

p =  0.75

pr(rej) =  0.5012019230769231

Tq =  0.039453454784761886

fixed Ts =  0.015

mean Tw =  0.0046793073746107935

q =  0.9839653687305397

w =  0.2339653687305397

--------------------------------
```

Part(e)

For $K = 3$, $\lambda = 65$, T_s is fixed and $= 0.015$, and simulation time $= 100$...

```
--------------------------------
     M/D/1/K Queue Analysis
--------------------------------

p =  0.975

pr(rej) =  0.49649532710280375

Tq =  0.04020818537115611

fixed Ts =  0.015

mean Tw =  0.005245009223093791

q =  1.3159255995010963

w =  0.3409255995010964

--------------------------------
```