CSE 538 – Fall 2015 Midterm
4 Problems – 80 points total

Your Name:

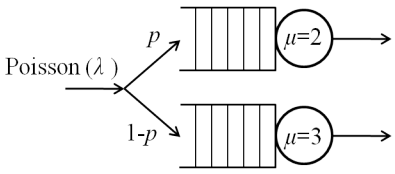

Figure 1: Open network model.

Problem 1 [15 points] Consider the open network of Fig. 1 with two servers of rates $\mu_1 = 2$ and $\mu_2 = 3$. Jobs arrive according to a Poisson process of rate λ, with jobs assigned to server 1 with probability p and to server 2 with probability $(1 - p)$.

1. **[5 points]** Assuming a stable system and a corresponding λ value, give the system's maximum throughput and the value(s) of p for which it is realized.

   In a stable open system, the throughput is constant and equal to the arrival rate λ. This is independent of the value chosen for p, as long as stability is ensured.

2. **[5 points]** What is the value of p that maximizes the system's stability region, *i.e.,* will allow the highest possible value of λ while ensuring stability? Justify your answer.

   Stability is ensured as long as we have

   $$\lambda p < \mu_1 = 2 \quad \text{and} \quad \lambda(1-p) < \mu_2 = 3$$

   This implies

   $$\lambda < \min\left\{\frac{2}{p}, \frac{3}{1-p}\right\}$$

   From the above expression, we see that λ is maximized when $\frac{2}{p} = \frac{3}{1-p}$. In other words, $p = \frac{2}{5}$.

3. **[5 points]** Assume that $\lambda = 2$ and compute, as a function of p, the probability $P\{3, 0\}$ that there are three (3) jobs in the top system **and** that the bottom system is empty (0 jobs). Explain what happens when $p = 1$.

   Each system behaves as an independent M/M/1 queue with loads $\rho_1 = \frac{2p}{2} = p$ and $\rho_2 = \frac{2(1-p)}{3}$. Hence, we know that the probabilities of $i \geq 0$ jobs in each systems are of the form

   $$\pi_i^{(1)} = \rho_1^i(1-\rho_1) = p^i(1-p) \quad \text{and} \quad \pi_i^{(2)} = \rho_2^i(1-\rho_2) = \left(\frac{2(1-p)}{3}\right)^i \frac{1+2p}{3}$$

   The probability $P\{3, 0\}$ that there are three jobs in the top system (system 1) and that the bottom system (system 2) is empty is, therefore equal to $\pi_3^{(1)} \times \pi_0^{(2)}$, *i.e.,*

   $$P\{3,0\} = (1-p)p^3\frac{(1+2p)}{3}$$

   When $p = 1$, no jobs are sent to the second system so that $\pi_0^{(2)} = 1$, but at the same time the first system becomes unstable. The latter implies that the first queue never stabilizes and therefore $\pi_i^{(1)} = 0, \forall\, i$, which implies that $P\{3, 0\} = 0$. .

Problem 2 [35 points] Consider the interactive system of Fig. 2 that consists of a CPU and two disks, a fast one and a slow one. Jobs visit the CPU once, but can visit the disks multiple times (multiple R/W operations). Note that p may be a design parameter, *i.e.,* we may be able to influence what fraction of R/W operations go to the fast disk, but β is outside our control, *i.e.,* is a function of the data footprint of the instructions being executed.

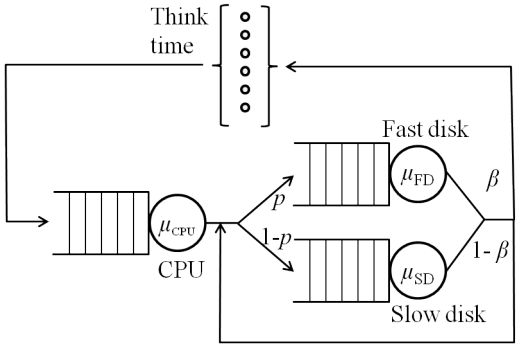

Figure 2: Closed system model.

We make the following measurements to assess the system's performance:

- Measurement duration: 20 minutes
- Average think time: 10 seconds
- Number of completed transactions in measurement interval: 1,500
- Number of CPU visits: 1,500
- Number of fast disk accesses: 30,000
- Number of slow disk accesses: 10,000
- CPU busy time: 1,000 seconds
- Fast disk busy time: 500 seconds
- Slow disk busy time: 600 seconds

1. **[10 points]** What is the average total service time of an individual transaction?

   We first compute the average service times of an individual visit to the CPU, fast disk and slow disk. Specifically, we have

$$\begin{aligned} S_{\text{CPU}} &= \frac{1000}{1500} = \frac{2}{3} \text{ seconds} \\ S_{\text{fast\_d}} &= \frac{500}{30000} = \frac{1}{60} \text{ seconds} \\ S_{\text{slow\_d}} &= \frac{600}{10000} = 0.06 \text{ seconds} \end{aligned}$$

   Similarly, we can obtain from the measurements the average number of visits to the CPU, fast disk, and slow disk per completed transactions.

$$\begin{aligned} V_{\text{CPU}} &= \frac{1500}{1500} = 1 \text{ visit} \\ V_{\text{fast\_d}} &= \frac{30000}{1500} = 20 \text{ visits} \\ V_{\text{slow\_d}} &= \frac{10000}{1500} = \frac{20}{3} \text{ visits} \end{aligned}$$

Using the fact that $D = V \cdot S$, we get

$$\begin{aligned} D_{\text{CPU}} &= \frac{2}{3} \text{ second} \\ D_{\text{fast\_d}} &= \frac{20}{60} = \frac{1}{3} \text{ second} \\ D_{\text{slow\_d}} &= \frac{20}{3} \times 0.06 = 0.4 \text{ second} \end{aligned}$$

so that the average total service time of a transaction is equal to

$$D = D_{\text{CPU}} + D_{\text{fast\_d}} + D_{\text{slow\_d}} = \frac{2}{3} + \frac{1}{3} + 0.4 = 1.4 \text{ seconds}$$

2. **[5 points]** Give asymptotic bounds for the system's throughput X and response time $E[R]$, as a function of N, the number of terminals using the interactive system.

   From the above expression for D, we get

$$D_{\max} = \frac{2}{3} \text{ second}$$

   Since we also know that the average think time $Z = 10$ seconds, we have

$$\begin{aligned} X &\leq \min\left(\frac{N}{11.4}, 1.5\right) \\ E[R] &\geq \max(1.4, N \cdot \frac{2}{3} - 10) \end{aligned}$$

3. **[20 points]** We are considering making the following changes to the system.

   (a) Turn the slow disk off;
   (b) Add a second fast disk and load-balance across all three disks;
   (c) Replace the CPU by one that is 50% faster;
   (d) All of the above, *i.e.,* faster CPU, slow disk off and load-balancing across two fast disks.

   Provide expressions, function of N, for the system's throughput X and response time $E[R]$ in each of the four configurations **[5 points each]**.

   (a) Turning the slow disk off means that the fast disk now gets to handle on average $V_{\text{slow\_d}}$ additional visits for each transaction. In other words,

$$V_{\text{fast\_d,(a)}} = V_{\text{fast\_d}} + V_{\text{slow\_d}} = 20 + \frac{20}{3} = \frac{80}{3}$$

   Hence, the new average service time per transaction for the fast disk is

$$D_{\text{fast\_d,(a)}} = \frac{80}{3} \cdot \frac{1}{60} = \frac{4}{9} \text{ second}$$

This does not change $D_{\max}$ since the CPU remains the bottleneck, but it affects D. Specifically, we have

$$D_{\text{(a)}} = \frac{2}{3} + \frac{4}{9} = \frac{10}{9} = 1.111 \text{ seconds}$$

This then yields

$$\begin{aligned} X_{\text{(a)}} &\leq \min\left(\frac{N}{11.11}, 1.5\right) \\ E[R_{\text{(a)}}] &\geq \max(1.11, N \cdot \frac{2}{3} - 10) \end{aligned}$$

(b) We now add a fast disk and rebalance the load across all three disks. This means that we want $D^{(1)}_{\text{fast\_d},(b)} = D^{(2)}_{\text{fast\_d,(b)}} = D_{\text{slow\_d,(b)}}$, while keeping the total average number of disk visits per transaction constant and equal to $\frac{80}{3}$. Using the fact that by symmetry we have $V^{(1)}_{\text{fast\_d,(b)}} = V^{(2)}_{\text{fast\_d,(b)}}$, this gives the following set of equations

$$\begin{aligned} \frac{V_{\text{fast\_d,(b)}}}{60} &= V_{\text{slow\_d,(b)}} \cdot 0.06 \\ 2V_{\text{fast\_d,(b)}} + V_{\text{slow\_d,(b)}} &= \frac{80}{3} \end{aligned}$$

After some manipulations, this yields $V_{\text{slow\_d,(b)}} \approx 3.25$ and $V_{\text{fast\_d,(b)}} \approx 11.707$, and therefore

$$D^{(1)}_{\text{fast\_d,(b)}} = D^{(2)}_{\text{fast\_d,(b)}} = D_{\text{slow\_d,(b)}} = 0.195 \text{ second}$$

and therefore $D_{\text{(b)}} = \frac{2}{3} + 3 \times 0.195 = 1.252$ seconds. Note that adding a second fast disk without turning off the slow disk performs worst than a single fast disk. This because the visits directed to the slow disk lower the efficiency of the disk sub-system. In this scenario as in the previous one, the CPU remains the bottleneck, so that $D_{\max}$ is unchanged and we have

$$\begin{aligned} X_{\text{(b)}} &\leq \min\left(\frac{N}{11.252}, 1.5\right) \\ E[R_{\text{(b)}}] &\geq \max(1.252, N \cdot \frac{2}{3} - 10) \end{aligned}$$

(c) Replacing the CPU by one that is 50% faster means that we now have $D_{\text{CPU,(c)}} = \frac{2}{3}/\frac{3}{2} = \frac{4}{9}$ second. The CPU, however, remains the bottleneck so that we have

$$\begin{aligned} D_{\max,\text{(c)}} &= \frac{4}{9} = 0.444 \text{ second} \\ D_{\text{(c)}} &= \frac{4}{9} + \frac{1}{3} + 0.4 = 1.178 \text{ seconds} \end{aligned}$$

As a result, we get

$$\begin{aligned} X_{\text{(c)}} &\leq \min\left(\frac{N}{11.178}, 2.25\right) \\ E[R_{\text{(c)}}] &\geq \max(1.178, N \cdot \frac{4}{9} - 10) \end{aligned}$$

(d) If we now consider the scenario where we have a faster CPU and two fast disks across which we load-balance, the total average service times of the two fast disks must satisfy

$$D_{\text{fast\_d,(d)}} = \frac{1}{60} \cdot \frac{80}{3} \cdot \frac{1}{2} = \frac{2}{9} = 0.222 \text{ second}$$

Since, as per the previous question, $D_{\text{CPU,(d)}} = \frac{4}{9} = 0.444$ second, we still have $D_{\text{max,(d)}} = D_{\text{CPU,(d)}} = 0.444$ second, while the total average service time per transaction is now equal to

$$D_{\text{(d)}} = \frac{4}{9} + \frac{2}{9} + \frac{2}{9} = \frac{8}{9} = 0.888 \text{ second}$$

This gives

$$\begin{aligned} X_{\text{(d)}} &\leq \min\left(\frac{N}{10.888}, 2.25\right) \\ E[R_{\text{(d)}}] &\geq \max(0.888, N \cdot \frac{4}{9} - 10) \end{aligned}$$

Problem 3 [20 points] Consider the two priority system of Fig. 3. High priority jobs arrive according to a Poisson process of rate λ_H, while low priority jobs arrive according to an independent Poisson process of rate λ_L. High and low priority jobs are assigned to separate queues, both of infinite capacity, but share a common server of unit service rate. Both high and low priority jobs have exponentially distributed service times with the same mean $1/\mu$. The system is assumed to be stable, *i.e.,* $\frac{\lambda_H+\lambda_L}{\mu} < 1$.

The system operates according to a *preemptive* priority policy. In other words, the server only serves jobs from the low priority queue if the high priority queue is empty. In particular, if a high priority job arrives (to an empty high priority queue) and finds the server busy serving a low priority job, the service of the low priority job is immediately interrupted and the server begins serving the high priority job. The low priority job resumes service only once the high priority queue is empty.

Note 1: The memoryless property of the exponential distribution ensures that a low priority job that resumes service is indistinguishable from one that just starts service.

Note 2: None of the questions below require solving for the probability distribution of the Markov chain representing the overall system.

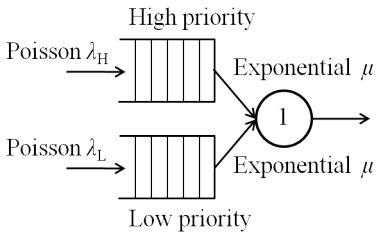

Figure 3: Priority system.

1. **[5 points]** Find an expression, function of the system parameters, for the probability $P^{(L)}_{\text{server busy}}$ that the server is busy serving a *low priority* job.

   We simply apply Little's Law to the server using the fact that the probability that the server is busy serving a low priority job is equal to the average number of low priority jobs in the server. In other words, we directly have

   $$P^{(L)}_{\text{server busy}} = E\left[N^{(L)}_{\text{server}}\right] = \frac{\lambda_L}{\mu} = \rho_L$$

2. **[5 points]** Find an expression, function of the system parameters, for the probability π_0 that the system is empty (both queues are empty and the server is idle).

   Applying again Little's Law to the server, we find that the probability that the server is busy (the average number of customers in the system) is given by

   $$\begin{aligned} P\{\text{server busy}\} &= E[N_{\text{server}}] = (\lambda_L+\lambda_H)\cdot\left[\frac{\lambda_L}{\lambda_L+\lambda_H}\cdot\frac{1}{\mu}+\frac{\lambda_H}{\lambda_L+\lambda_H}\cdot\frac{1}{\mu}\right] \\ &= \frac{\lambda_L}{\mu}+\frac{\lambda_H}{\mu} = \rho_H+\rho_L = \rho \end{aligned}$$

   Alternatively, we could have used the reasoning of the previous question and applied it to high priority jobs to find that the server was busy serving high priority jobs with probability $\frac{\lambda_H}{\mu}$. The probability that the server is busy is then simply the sum of the probabilities that it is busy serving a high or low priority job. Hence, the probability π_0 that the system is empty is given by

   $$\pi_0 = 1 - P\{\text{server busy}\} = 1 - \left[\frac{\lambda_L}{\mu}+\frac{\lambda_H}{\mu}\right] = 1-\rho$$

3. **[5 points]** Find an expression, function of the system parameters, for the average number of high priority jobs in the system, $E[N_H]$.

   Because the system operates according to a preemptive resume priority policy, the low priority jobs are essentially transparent to the high priority jobs. Hence, the high priority queue behaves like a regular M/M/1 system with arrival rate λ_H and service rate μ. This implies that

$$E[N_H] = \frac{\rho_H}{1-\rho_H}, \quad \text{where} \quad \rho_H = \frac{\lambda_H}{\mu}$$

4. **[5 points]** Now, find an expression, function of the system parameters, for the average number of low priority jobs in the system, $E[N_L]$. (**Hint**: Unlike the result of the previous question that can be derived directly, this requires an intermediate step.)

   We know that $E[N] = E[N_H] + E[N_L]$ with Little's Law applied to the entire system giving us $E[N] = \frac{\rho}{1-\rho}$, where $\rho = \rho_H + \rho_L$. Combining this with the result of the previous question gives

$$\begin{aligned} E[N_L] &= E[N] - E[N_H] = \frac{\rho_H + \rho_L}{1-\rho_H-\rho_L} - \frac{\rho_H}{1-\rho_H} \\ &= \frac{\rho_L}{(1-\rho_H-\rho_L)(1-\rho_H)} \end{aligned}$$

Problem 4 [10 points] Consider the discrete time Markov chain (DTMC) of Fig. 4.

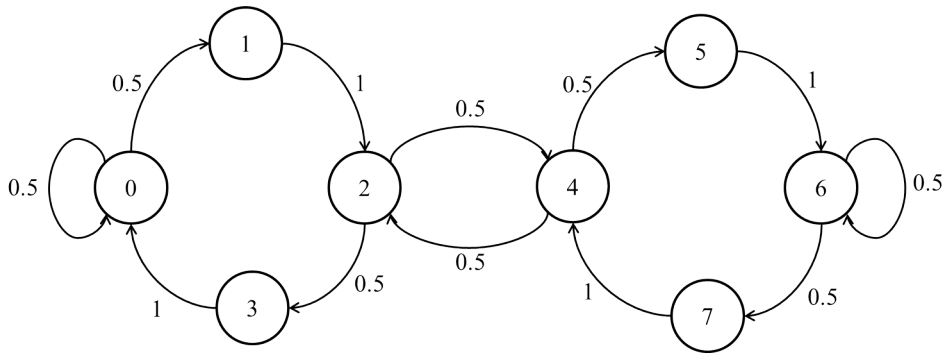

Figure 4: Discrete time Markov chain.

1. **[2 points]** Does the chain admit a limiting distribution? Justify your answer.

   The chain is finite, irreducible and aperiodic (because it has self-loops), and therefore it admits a limiting distribution.

2. **[8 points]** Compute the stationary probabilities $\pi_0, \pi_1, \ldots, \pi_7$ for the chain.

   The chain's balance equations give

$$\begin{aligned} \pi_2 &= \pi_1 + 0.5\pi_4 & \qquad \pi_4 &= \pi_7 + 0.5\pi_2 \\ \pi_3 &= 0.5\pi_2 & \qquad \pi_5 &= 0.5\pi_4 \\ 0.5\pi_0 &= \pi_3 & \qquad 0.5\pi_6 &= \pi_5 \\ \pi_1 &= 0.5\pi_0 & \qquad \pi_7 &= 0.5\pi_6 \end{aligned}$$

   which can be readily seen to imply

$$\begin{aligned} \pi_2 &= \pi_2 & \qquad \pi_4 &= \pi_2 \\ \pi_3 &= 0.5\pi_2 & \qquad \pi_5 &= 0.5\pi_2 \\ \pi_0 &= \pi_2 & \qquad \pi_6 &= \pi_2 \\ \pi_1 &= 0.5\pi_2 & \qquad \pi_7 &= 0.5\pi_2 \end{aligned}$$

   which together with the normalization equation $\sum_{i=0}^{7} \pi_i = 1$ gives

$$\begin{aligned} \pi_2 &= \frac{1}{6} & \qquad \pi_4 &= \frac{1}{6} \\ \pi_3 &= \frac{1}{12} & \qquad \pi_5 &= \frac{1}{12} \\ \pi_0 &= \frac{1}{6} & \qquad \pi_6 &= \frac{1}{6} \\ \pi_1 &= \frac{1}{12} & \qquad \pi_7 &= \frac{1}{12} \end{aligned}$$