

Elec 428
Final Exam Solutions

1. (a) A switch in a computer communications network has eight input links and eight output links. Messages arrive over each input link at rate λ messages per second (a Poisson process). Under normal circumstances, each message that arrives at the switch is routed to output links 1-7 with probability 0.1 each and to output link 8 with probability 0.3. A link can transmit only one message at a time. The time to transmit a message over an output link is exponentially distributed with mean $4/(5\lambda)$.

What is the average number of messages in each output link message queue (including any messages that are in the process of being transmitted on the link)? (10 pts)

The arrival rate λ_{1-7} at each of the output links 1 – 7 is $\lambda_{1-7} = 8\lambda(0.1) = 0.8\lambda$. The arrival rate λ_8 at output link 8 is $\lambda_8 = 8\lambda(0.3) = 2.4\lambda$. The utilizations of the output links 1 – 7 are

$$\rho_{1-7} = \lambda_{1-7}/\mu = 0.8\lambda/1.25\lambda = 0.64$$

A similar calculation for output link 8 gives

$$\rho_8 = \lambda_8/\mu = 2.4\lambda/1.25\lambda = 1.92 > 1$$

Hence, the queue representing output link 8 is not stable (not ergodic) and has no steady-state average queue length.

- (b) To relieve the congestion on link 8, the switch employs a congestion control algorithm. If the number of messages in the queue of output link 8 (including the message being transmitted) is four, a message that normally would be routed to output link 8 is routed to one of the other seven output links, with equal probability for each link. What is the average number of messages in output link 8's queue? (10 pts)

Since the queue length can be no greater than 4, the queue is stable, and we can solve it as a birth-and-death process, because the arrivals to output link 8 constitute a Poisson process and the service (transmission) time for the link is exponentially distributed. Using the local balance equations, we get

$$\pi_1 = \rho_8\pi_0$$

$$\pi_2 = \rho_8^2\pi_0$$

$$\pi_3 = \rho_8^3\pi_0$$

$$\pi_4 = \rho_8^4\pi_0$$

Substituting these into the normalization equations gives

$$\begin{aligned}
 1 &= \pi_0(1 + \rho_8 + \rho_8^2 + \rho_8^3 + \rho_8^4) \\
 &= \pi_0 \left(\frac{1 - \rho_8^5}{1 - \rho_8} \right) = \pi_0 \left(\frac{1 - 1.92^5}{1 - 1.92} \right) = \pi_0(27.273833)
 \end{aligned}$$

$$\Rightarrow \pi_0 \approx 0.036665$$

The average number of messages in the output link 8 queue is

$$\begin{aligned}
 \bar{N}_8 &= 1 \cdot \pi_1 + 2 \cdot \pi_2 + 3 \cdot \pi_3 + 4 \cdot \pi_4 \\
 &\approx (84.8846)(0.036665) \approx 3.11231
 \end{aligned}$$

- (c) What is the approximate average number of messages in each of the other output link queues when the congestion control algorithm of part (b) is used? (5 pts)

All arrivals when the output link 8 has four messages queued are redirected to one of the other seven links with equal probability. The arrival rate at links 1 – 7 is

$$\begin{aligned}
 \lambda_{1-7}^{\circledast} &= \lambda_{1-7} + (\lambda_8 \pi_4) / 7 \\
 &= 0.8\lambda + (2.4\lambda \cdot 0.498263112) / 7 \approx 0.97\lambda
 \end{aligned}$$

The arrival processes at output links 1 – 7 are no longer independent, homogeneous Poisson processes, since the arrival rate will depend on the number of messages in output link 8's queue. However, we can make the simplifying assumption that the input processes are Poisson with rate 0.97λ . This allows us to use an M/M/1 model for output links 1 – 7.

$$\rho_{1-7}^{\circledast} = \lambda_{1-7}^{\circledast} / \mu = 0.97\lambda / 1.25\lambda \approx 0.7767$$

For the M/M/1 model, the average number of messages in the queue is

$$\bar{N}^{\circledast} = \frac{\rho_{1-7}^{\circledast}}{1 - \rho_{1-7}^{\circledast}} \approx 3.4783$$

2. (a) A computer system consists of a CPU and two independent disks. The system sees two types of jobs. The first type is a batch-type job. Each of these jobs repeats a cycle in which it first receives service from the CPU and then receives service from one of the two disks. The probability that a job chooses disk 1 during any cycle is 0.6, independent of the choice it makes in any other cycle. The average service time at the CPU on each visit is 0.005 seconds; the average service times per visit at disk 1 and disk 2 are 0.03 and 0.07 seconds, respectively. At any given time, there are four of these jobs in the system.

The second type of job is a real-time control application. At random moments in time, a sensor triggers a request to the CPU to perform a computation and

send a response to a device. Each such event requires only CPU service; no disk activity is involved. Because the triggering event requires a real-time response, these computations preempt the batch computations. The trigger events occur 30 times per second on average; the average time the CPU takes to compute the response to an event is 0.03 seconds, and the standard deviation is also 0.03 seconds.

What is the CPU throughput in total jobs (batch and real-time) per second? What is the average response time for a real-time job (waiting time plus service time)? (15 pts)

This is a mixed-class queueing network that is not product-form because of the preemptive priority given to real-time jobs. However, we can assume that real-time events constitute a Poisson process and get the average response time for real-time jobs by treating the CPU as an M/G/1 queue.

$$\bar{T} = \bar{X} + \frac{\lambda \bar{X}^2}{2(1-\rho)} = 0.03 + \frac{30(0.0018)}{2(.1)} = 0.3 \text{ seconds}$$

Also, since the utilization of the CPU by real-time jobs can be determined, we can get the throughput of the batch-type jobs by inflating their CPU service demands to account for the CPU utilization by real-time jobs, just like in the mixed-class MVA algorithm.

$$\rho_{RT,CPU} = \lambda_{RT} \cdot \bar{X}_{RT} = 30(0.03) = 0.9$$

$$D_{RT,CPU}^* = \frac{0.005}{1-0.9} = 0.05$$

$$D_{RT,d1}^* = D_{RT,d1} = 0.6(0.03) = 0.018$$

$$D_{RT,d2}^* = D_{RT,d2} = 0.4(0.07) = 0.028$$

since real-time jobs do not visit the disks. Applying single-class MVA for just the closed system consisting of batch-type jobs:

	N = 0	N = 1	N = 2	N = 3	N = 4
R_{CPU}		0.05	0.07604	0.1069	0.1423
R_{d1}		0.018	0.02137	0.0238	0.0254
R_{d2}		0.028	0.03415	0.0432	0.0489
X		10.1467	14.9719	17.2566	18.4763
Q_{CPU}	0	0.5208	1.1385	1.8452	2.62918
Q_{d1}	0	0.1875	0.3200	0.4100	
Q_{d2}	0	0.2197	0.5415	0.7448	

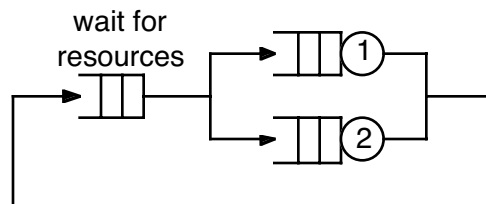
The total throughput for the CPU is $18.4763 + 30 = 48.4763$ jobs per second.

- (b) If the real-time jobs do not preempt the batch jobs but are subject to the same scheduling policy as the batch jobs, what is the average response time of a real-time job? (10 pts)

Without preemption, this becomes a standard mixed-class product-form network. The response time, throughput, and queue length for batch jobs at the CPU is computed the same as in part (a). The response time for real-time jobs is

$$R_{RT,CPU} = \frac{D_{RT,CPU}}{1 - \rho_{RT,CPU}} (1 + Q_{B,CPU}) \approx \frac{0.3}{1 - 0.9} (1 + 2.62918) \approx 1.08875$$

3. A system has two queues, numbered 1 and 2, as shown in the following figure.



It also contains 2 jobs of class A and 2 jobs of class B. Class A jobs may visit either queue 1 or queue 2, while class B jobs only visit queue 2. The visit ratios and per visit demands of the two job classes for the two queues are:

$$\begin{aligned} V_{A,1} &= 2/3 & V_{A,2} &= 1/3 & V_{B,2} &= 1 \\ \bar{X}_{A,1} &= 15 & \bar{X}_{A,2} &= 15 & \bar{X}_{B,2} &= 15 \end{aligned}$$

The system is complicated by the presence and use of two pools of resources, Y and Z, each containing four units. In order to receive service at either queue 1 or queue 2, a class A job needs two units of resource Y and one unit of resource Z. A class B job must obtain one unit of resource Y and two units of resource Z before it can receive service at queue 2.

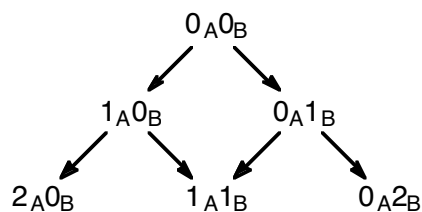
The system has a waiting area (cleverly labeled “wait for resources” in the figure) where jobs wait, in FCFS order, for the resources they need before they can proceed to a queue. The job at the head of the waiting area queue waits until there are enough free units of both types of resources, claims the required units of resources (removes them from their respective pools), and proceeds to a queue for service. After completing service at the queue, the job returns all units of the resources it holds to the pools and joins the “wait for resources” queue at the end. Note that the “wait for resources” queue is simply a waiting area; jobs do not receive service there.

What is the approximate throughput for class A jobs? (25 pts)

This network is not product-form, because of the need for possession of the resources. We can find an approximate solution using a FESC. First, divide the system into the complement, consisting of the “wait for resources” area, and the aggregate, made up of the two queues with the “wait for resources” area replaced by a short. Compute the demands for class A and B jobs. Throughput is measured along the short.

$$\begin{aligned} D_{A,1} &= 10 & D_{B,1} &= 0 \\ D_{A,2} &= 5 & D_{B,2} &= 5 \end{aligned}$$

The need for resources constrains the feasible populations for the aggregate: no more than two jobs, regardless of class, may be in the aggregate at one time. Hence, the precedence graph for feasible populations is



Use multiple-class, closed system MVA to find the throughputs for the aggregate’s feasible populations.

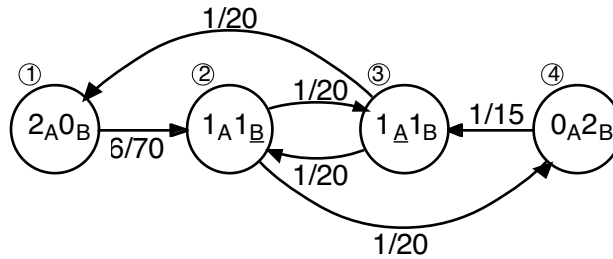
	0_A0_B	1_A0_B	0_A1_B	2_A0_B	1_A1_B	0_A2_B
$R_{A,1}$		10		50/3	10	
$R_{A,2}$		5		20/3	10	
$R_{B,2}$			15		20	30
X_A		1/15		6/70	1/20	
X_B			1/15		1/20	1/15
$Q_{A,1}$	0	2/3	0	1.4286	1/2	
$Q_{A,2}$	0	1/3	0	0.5714	1/2	
$Q_{B,2}$	0	0	1	0	1	2

Now replace the aggregate in the original model by a FESC with load-dependent service rates, and solve using a continuous-time Markov chain. The Markov chain does not, however, require a state for each of the feasible populations in the above MVA table. The first three columns deal with feasible populations that cannot exist in the aggregate in steady-state, since in each case there would always be enough resources left to allow whatever job was at the head of the “wait for resources” area queue to proceed.

We might first try a Markov chain with three states: (2_A0_B) , (1_A1_B) , and (0_A2_B) . The load-dependent service rates for the aggregate in each state are the throughputs for the corresponding feasible populations in the above table. Also, from state (2_A0_B) and state (0_A2_B) the system always goes to state (1_A1_B) , because jobs queue in the waiting area in FCFS order. However, in state (1_A1_B) , the next state depends on the class of

the job at the head of the queue in the waiting area. Consequently, we need to represent this feasible population by two states: one corresponding to a class A job at the head of the waiting area queue, and the other to a class B job at the head.

The resulting Markov chain has four states, and is described by the following state diagram:



In state (1_A1_B), a class B job is at the head of the waiting area queue. In state (1_A1_B), class A job is first in line. The transition rate matrix is

$$Q = \begin{bmatrix} -\frac{6}{70} & \frac{6}{70} & 0 & 0 \\ 0 & -\frac{1}{10} & \frac{1}{20} & \frac{1}{20} \\ \frac{1}{20} & \frac{1}{20} & -\frac{1}{10} & 0 \\ 0 & 0 & \frac{1}{15} & -\frac{1}{15} \end{bmatrix}$$

(the states are listed in the same left-to-right order in which they appear in the state diagram). Solving $\underline{\pi}Q = \underline{0}$ with the help of the normalization equation gives

$$\underline{\pi} = \left(\frac{7}{40} \quad \frac{12}{40} \quad \frac{12}{40} \quad \frac{9}{40} \right)$$

The approximate throughput for class A jobs is $\pi_1 X_A(2_A 0_B) + (\pi_2 + \pi_3) X_A(1_A 1_B) = 9/200$.

- A queue has a server that consists of a sequence of 10 stages. The service time in each stage is exponentially distributed with rate 10. A job entering service begins at the first stage, proceeds in turn through all subsequent stages, and exits the server. While it is in any stage, no other job may be begin service. Jobs arrive at the queue according to a Poisson process with rate 0.5 jobs per second. What is the average response time (waiting time plus service time) of a job in steady state? (25 pts)

This is a simple 10-stage Erlang server. The average service time in each stage is 1/10 second. Since each stage has an exponentially distributed service time, the standard deviation in each stage is also 1/10 second. The mean service time for the entire server is $10 \cdot 1/10 = 1$ second, and the variance for the entire server is $10 \cdot (1/10)^2 = 1/10 \text{ sec}^2$. The second principal moment of the service time distribution is $\overline{X^2} = \sigma_X^2 + \overline{X}^2 = 1/10 + 1 = 11/10 \text{ sec}^2$. Apply the Pollaczek-Khinchine mean value formula to obtain

$$\bar{T} = \bar{X} + \frac{\lambda \bar{X}^2}{2(1-\rho)} = 1 + \frac{0.5(11/10)}{2(1-0.5 \cdot 1)} = \frac{31}{20} \text{ sec.}$$