

1. (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)?

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?

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

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