ELEC-E7460 Modelling & Simulation

Mathematica Assignment 3

Lou Zhi Qi
726533 zhi.lou@aalto.fi

Problem 1

(a) Values obtained from resubmission Q1a simulation were:

Policy	Overall Mean Delay
Randomized Load Balancing	1.65506
JSQ	1.24611
WJSQ	1.16647

(b) Values obtained from resubmission Q1b simulation were:

	RandLoadBalance		JSQ		WJSQ	
λ	Mean	Confidence Interval	Mean	Confidence Interval	Mean	Confidence Interval
0.5	0.44613	(0.443818, 0.448443)	0.777243	(0.775951, 0.778535)	0.758364	(0.756166, 0.760562)
1.5	0.570583	(0.568407, 0.572759)	0.694445	(0.693269, 0.695621)	0.616348	(0.615223, 0.617472)
2.5	0.802783	(0.798325, 0.807241)	0.790573	(0.788926, 0.79222)	0.650562	(0.648773, 0.65235)
3.0	0.999996	(0.995443, 1.00455)	0.903457	(0.900996, 0.905919)	0.728035	(0.725519, 0.730551)
3.5	1.33689	(1.32722, 1.34656)	1.10198	(1.097, 1.10697)	0.881706	(0.877811, 0.885602)
4.0	1.99744	(1.97507, 2.0198)	1.48568	(1.47481, 1.49654)	1.2082	(1.19963, 1.21676)

From the results, `Random Load Balancing` seems to perform much better than the other two policies when load is small ($\rho = 0.1, 0.3$), and much worse when load is higher. Between `JSQ` and `WJSQ`, the latter consistently performed better than JSQ. This is likely due to the fact that WJSQ takes into consideration the queues' respective service rate, which would affect the rate of job departures and hence delay experienced by jobs in the respective queues. Overall, I would say that `WJSQ` performed the best. Nevertheless, `Random Load Balancing` would still be the more appropriate policy to use when load is known to be small.

The difference in performance across different values of ρ is affected by the longer queues. As load increases, arrival rate increases and queue lengths increase.