COMP 5320/6320/6326
Design and Analysis of Computer Networks

Term Project
Assigned on 11/12/2018
Due by 12/07/2018

Important Notice: This is a team-based project. Each team should consist of no more than 2 students. Team members of the same team will receive the same score. I strongly suggest you to team up with the same team member as in your previous lab assignment. Each team should finish its project independently. Your final submission should include a written project report, all source code of your implementation, and executables compiled from the source code, all zipped into a single file named project_GID.zip (or project_GID.rar), where GID is the group ID of your team (the same as in your previous lab assignment). In addition to submitting this file to Canvas, you should also submit a printed version of your report in the class on 12/07. Each team should only submit one copy, for both the file and the printed report.

Project Description

In this project, we will evaluate by simulation the performance of a two-queue system as shown in Figure 1, under various packet-assignment strategies. Each queue in the system has a capacity of 10 packets (including the one in the server). A packet-assignment strategy dictates how an incoming packet is assigned to one of the queues. We are interested in the following two strategies: (1) Random selection, which assigns an incoming packet uniformly randomly to one of the queues; (2) min-queue strategy, which assigns the packet to the queue of the minimum length. For each queue we assume that FCFS service policy is used.

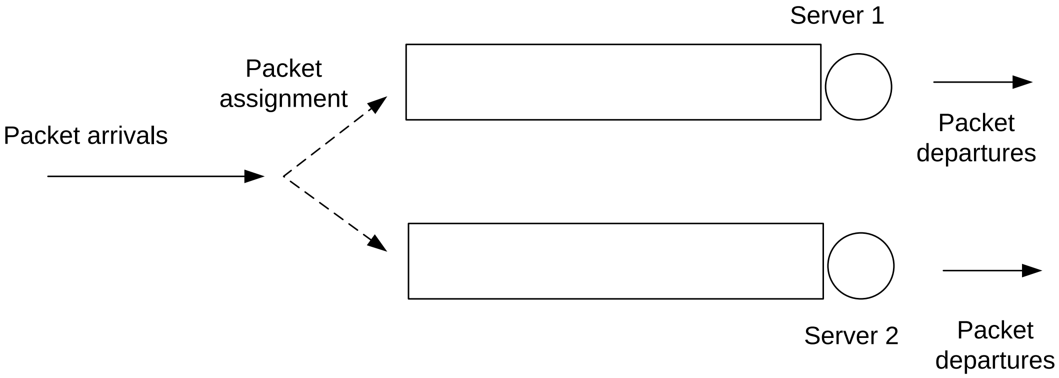

Figure 1. A two-queue switch

We are interested in the following three performance metrics:

1. Blocking probability of an incoming packet: This is defined as the ratio of the number of dropped packets (at both queues) to the total number of packets offered to the system.

2. Average queue length: This is defined as the average number of packets (including the one being served) in an arbitrary queue sampled at the arrival time of a packet.

3. Average waiting time: This is defined as the average time an admitted packets needs to stay inside the queue, i.e., the interval from its arrival to the moment when its service is finished. Note that different from the above two metrics that are calculated based on the entire set of offered packets, the average waiting time only applies to the packets that are admitted to one of the queues.

The above performance metrics will be evaluated against the following parameters.

1. Arrival rate λ: We assume that the packet arrivals follow a Poisson process of rate λ.

2. Service rate μ: We assume that the service time for a packet is exponentially distributed with mean 1/μ.

3. Traffic load ρ: For the two-queue system, ρ=λ/2μ.

Your tasks:

1. Through packet-level simulation, compare the performance of the two packet-assignment strategies against each of the parameters. Draw a performance figure for each comparison. In each figure, the Y-axis should be the performance metric being evaluated, and the X-axis is the range of parameter being considered. In each figure, please also plot the theoretical results for the random-selection strategy. So in total you should have 9 figures, and in each figure three curves are plotted (two are simulated and one is theoretical). To make your curve smooth, each data point should be based on the average of at least 10 independent runs (10 seeds), and each run should simulate 10,000 offered packets.

2. Sensitivity analysis: Through packet-level simulation, identify the key parameters that impact the performance most significantly.

3. Please show all results (figures) in your written report. You should also describe and explain each figure briefly. Especially you should explain why the simulated results are correct. For task 2, you should also describe your simulation setup that allows you to verify the sensitivity of performance to each parameter.

All simulations must be implemented in C. Your program should be compiled and executed on the Tux machines. Note that you may need to write more than one program. Matlab can be used only to plot figures.