Computer Engineering 4DK4
Lab 2
Integrated Packet Voice and Data Link Performance

This lab is an introduction to discrete-event simulation using the Simlib library. The provided simulation models a system where packets arrive at a packet switching node and are stored in a buffer, awaiting transmission on an output link. It is assumed that the packets are fixed in length and arrive according to a Poisson process. The system can therefore be modelled as a single server (M/D/1) queueing system as in the simulation from Lab 1. Mean delay versus throughput results are first obtained for this default case. Some experiments are then performed which determine the packet delay distribution performance of the node. The simulation is then modified to quantify the delay performance when a second transmission link is added to the node and operated concurrently with the first. Finally, an integrated voice/data switch is simulated and real-time voice performance comparisons are performed for two different packet queueing disciplines.

1 Preparation

1. It is important that you attend the lectures/tutorials which introduce the C program to be used. You will be responsible for knowing how the simulator works.

2. An electronic copy of the simulation program must first be obtained. The program consists of a number of C source (.c) and header (.h) files. You must also have the `simlib.h, simlib.c` and `trace.h` Simlib library files when you compile and link the simulation. A zip file of everything you need is available on the course web site.

3. You must compile and run the simulator. If you are not familiar with C and a C compiler make sure you see me or one of the TAs as soon as possible.

2 Experiments

1. Familiarize yourself with running the simulation program. The downloaded version simulates the same system as in Lab 1, i.e., an M/D/1 queueing system. Packets arrive to the buffer, awaiting transmission on the output link. When the simulation is running it writes output on the screen.

   There are various parameters such as the `PACKET_ARRIVAL_RATE`, `PACKET_LENGTH`, `LINK_BIT_RATE` and `RUNLENGTH`, which you can set in `simparameters.h`. There is also a `RANDOM_SEED_LIST` which consists of a comma separated list of random number generator seeds. When the simulation executes, it will do a separate simulation run for each seed. Once you are familiar with compiling and running the program, you can continue on and perform the following experiments.

2. Generate a plot of mean delay (in msec) vs `PACKET_ARRIVAL_RATE` in the same fashion as in Lab 1. Set `PACKET_LENGTH` to 1000 bits and assume that the output link operates at a bit rate of 1 Mbps (i.e., `LINK_BIT_RATE`). *Make sure that you include runs using your McMaster student ID number as the seed.*

3. Consider a packet switch where arrivals to be transmitted on an outgoing link can be modelled as a Poisson process with an arrival rate of λ packets/sec. When the link is busy transmitting, arriving packets are stored in a large buffer at the switch. The packets are drawn from the buffer in FCFS (first-come-first-served) order and transmitted non-preemptively, i.e., the transmission of a packet is always completed once it starts. Assume that all packets have a fixed length of 1000 bits each and that the output link operates at 1 Mbps. Modify the simulation and use it to find the maximum value of λ such that the probability that a packet's delay exceeds 20 msec is less than 2%.

4. Assume that we have a system as in Part 2 except that now there are two identical transmission links serving the same queue (This can be modelled as an M/D/2 queueing system). However, each of the links operates at 500 Kbps (i.e., half the bit rate compared to the single link in Part 2)[1]. The two links are always busy transmitting whenever packets are available, i.e., whenever either of the links becomes idle, a packet is taken into service from the front of the buffer, if one is available, and begins transmission. Modify the program to simulate this system. Make any other reasonable assumptions. Compare the mean delay versus throughput performance with that of the system in Part 2.

5. There is a packet switching link that is servicing both real-time voice and best-effort data traffic. The data traffic arrives according to a Poisson process as in Part 2, but where service times are exponentially distributed with a mean service time of 40 ms. There is also a voice traffic stream encoded using the G.711 (64 Kbps) voice codec. In this case, packets arrive periodically with fixed inter-packet arrival times equal to $t_v = 20$ ms. The packets include 62 bytes of header overhead in addition to the voice payload. Assume that the transmission link operates at 1 Mbps.

   Write a simulation for this situation assuming that all arriving packets are placed in the same buffer and served in FCFS order. The best way to do this is to add two new arrival events for the two voice streams. Generate results which show the mean delay of the voice and data packets versus the data packet arrival rate (i.e., `MEAN_ARRIVAL_RATE`). Plot a separate curve for each voice stream delay and data packet delay on the same graph.

6. In Part 5, now assume that voice packets are given (non-preemptive) priority over data packets, i.e., The server always serves voice packets first and data packets are only served when there are no waiting voice packets. However, once a data packet starts service, it is allowed to complete even if a new voice packet arrives. (This is the non-preemptive part.)

   Simulate this situation and compare the results you obtain with that found in Part 5. The easiest way to simulate this is to create separate voice and data packet queues. Then create separate arrival and departure events for both using the existing event functions as templates.

In all the experiments, explain the results that you obtain.

3 Writeup

Submit a writeup for the lab. Each group (of 2 maximum) is responsible for their own experiments and writeup. Include in your writeup a description of everything that you did including all data (and random number generator seeds) that were used to obtain the graphs. Include the plots and a listing of the modified program with your writeup.

[1] We can view it that about the same bandwidth is used in both cases.