# Computer Engineering 4DK4
# Lab 2
# Packet Switched Network and Integrated Voice Link Performance

This lab is an introduction to discrete-event simulation using the Simlib library. The provided C code 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 that determine the packet delay distribution performance of the packet switch. The simulation is then modified to quantify the mean delay performance of a network with three packet switches. 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 500 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.*

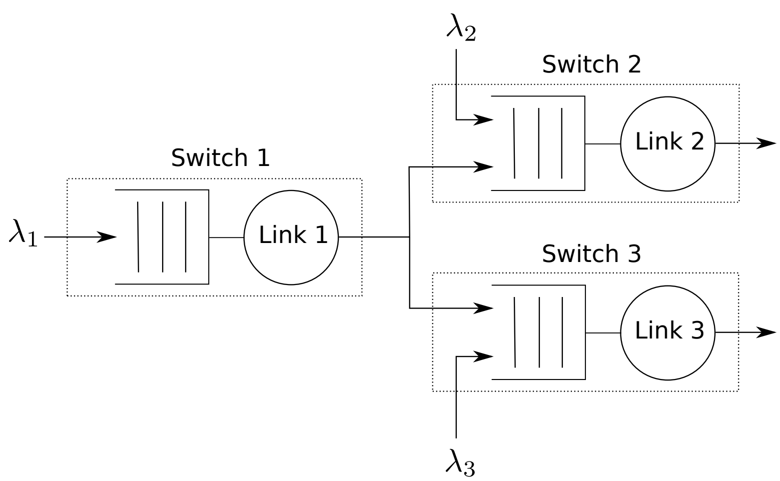
3. Consider a packet switch where arrivals to be transmitted on an outgoing link can be modelled as a Poisson process with an arrival rate of $\lambda$ packets/sec. When the link is busy transmitting, arriving packets are stored in a large buffer at the switch. The packets are drawn from the buffer in 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 2000 bits each and that the output link operates at 1 Mbps. Modify the simulation and use it to find the maximum value of $\lambda$ 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 are three packet switches connected into a network as shown below. Each of the switches has local packet arrivals at rates of $\lambda_1$, $\lambda_2$ and $\lambda_3$ packets/second, as shown. The packet arrival process can be modelled as Poisson and all packets are 1000 bits. At each switch, the arriving packets are buffered in FCFS order, awaiting transmission on the outgoing link.

   All of the packets that arrive at Switch 1 are transmitted over Link 1, which is a wireless link that Switches 2 and 3 are listening on. Each packet that is transmitted on Link 1 is destined for Link 2 with probability $p_{12} = 1 - p_{13}$, independently for each transmitted packet. Eventually, all arriving packets are transmitted over either Link 2 or 3. The transmission bit rates are 2 Mbps, 1 Mbps and 1 Mbps for Links 1, 2 and 3, respectively.

   Create a simulation that models the performance of this network. Fix $\lambda_2$ and $\lambda_3$ to the same value of 500 packets/second, and set $\lambda_1$ to 750 packets/second. Generate plots of mean delay vs. $p_{12}$, showing the mean delay experienced by packets originating at Switches 1, 2 and 3, on the same graph. The total delay of a packet is from the time it arrives to one of the switches, until it is fully transmitted on either Link 2 or 3.




6. 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 the service times are exponentially

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

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. That is, in every 20 ms period, all bits in the encoded voice traffic are packed into one voice packets. Each voice packet also includes 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 generate two new arrival events, one for voice traffic and another for data traffic. 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.

7. In Part 6, 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 6. 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.

1. Each group (of 2 maximum) should submit a single writeup.
2. Include in your writeup a description of everything that you did including all data and the random number generator seeds that were used to obtain the graphs.
3. Include with your writeup the plots and a listing of any of the code that you changed.
4. Unless otherwise indicated, your performance data and curves should cover the system operations from low traffic load to high traffic load.
5. No piecewise linear plots or overfitted curves are allowed.