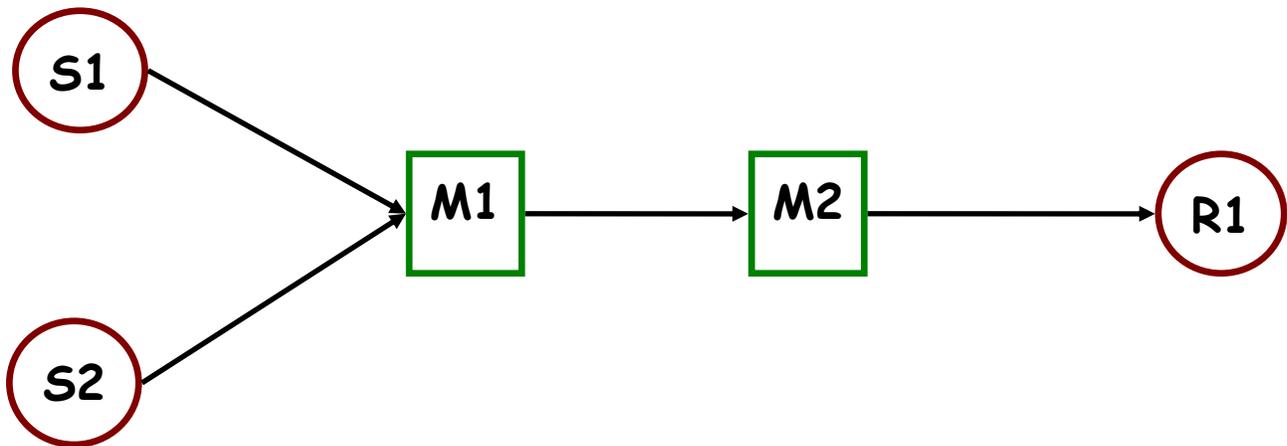


Simulating Packet Transmissions in C++



In Program 5, you must simulate MANET nodes. To assist the student in breaking up a large programming assignment into smaller design components, Lab 5 addresses an easier problem, simulating the transmission of network packets when the mule nodes are not moving.

For this simulation, time will be in simulated units (**sims**).

Assume you have two sending Host nodes (S1 and S2) sending packets to one receiver Host node (R1). Two intermediate nodes (M1 and M2) use FCFS packet queues. Both sending hosts transmit packets to intermediate node M1. M1 transmits packets to M2. M2 transmits packets to the final destination R1.

In this simulation all packets are the same size. All transmitting nodes (S1, S2, M1 and M2) wait 1 sim between sending packets to simulate an IPG (Inter Packet Gap).

Simulation Parameter Details

The following are internal details for this simple simulation:

S1 and S2 send 100 packets. S1 starts sending packets at time 0. S2 starts sending packets at time 2. The times needed for a packet to go from a transmitting node to the next node are:

S1 to M1	5 sims
S2 to M1	6 sims
M1 to M2	3 sims
M2 to R1	4 sims

This lab is designed as an opportunity to realize **code reuse**. The assumption is that you will be able to reuse your event list and FCFS queue implementations from Program 3. If you do not have a working version of Program 3, we will make a working version of Program 3 available early in the a.m. on Tuesday, February 12.

Main Assignment

This lab assignment runs over two labs with attending the second lab being optional. Your **C++ program** simulates the scenario detailed above. Your program should print out all packet response times for every packet that traverses the network (identifying the source) and the average packet response time for S1 and S2 packets. For bottleneck analysis also print out the maximum queue size experienced for the M1 and M2 queues.

Design Choice

Note, there are (at least) two ways to drive this simulation:

1. Start by putting all 200 packet transmission events on the event list before starting the simulation. This is acceptable for this lab but definitely 'clunky'. In Program 5 it is conceivable that a simulation involves sending **millions of packets** !!
2. Design the routine that processes the event simulating the arrival of a packet at M1 such that the arrival of packet **n** from Host S **triggers the insertion of a new event** in the event list for the arrival of packet **n+1** from Host S at M1. This is more complex, but the calculation of this future event is possible in this case because packets cannot be sent out of order and all packets arrive safely at the destination.

Lab 5 Assignment

0. Prior to coming to the lab prepare a preliminary solution to the program above.
 1. Create a make file.
 2. Note there is no test data for this lab.
 3. Run the program redirecting the output to `simplesim.out`.
 4. Create a README file that contains any useful information to assist in the grading of your lab program.
 5. Create a tarred file that contains all the source and header files, the make file and the README file and your output file.
 6. Use the Unix version of the 'turnin' to turn-in the tarred file. [The deadline for all lab turn-ins is 8 days after the beginning of your assigned lab 5.]