EE 122 – Project #2

Spring 2015

Total: 200 Points
Due: Friday, April 10, 2015, 11:59 pm
Submission: Please upload your code and write-up to bcourses

Complete this project in groups of two, using two laptops.

This project asks you to write some code and generate a write-up of results for experiments performed with this code. You will need to submit both the source code and the experimental results write-up to the bcourses website.

Step 1: Build a 2-Hop Network

You are required to build a 2-hop network with a router in the middle. Create a process modeling source S and one modeling for destination D and one modeling for a simple queuing model for the router L. You can use two computers, and put sender on one machine, receiver and router on another machine.

S communicates with D by sending UDP packets of 128 Bytes **of payload** using UDP sockets. You can adopt any strategy to create these packets, but each packet must carry at least a sequence number and a generation time-stamp as well as means to identify (at the UDP Application Level!) to which stream any packet belongs. Assume that the router has an internal storage allowing to store up to B = 64 packets. If the storage is full, arriving packets are discarded. Assume that the router forwards a packet from the front of the queue every 10ms (therefore, the service rate of L is **10ms/pkt**).

Let W be the inter-packet time, that is, the time duration between sending out packets by the sender S. This inter-packet time W follows Poisson distribution with mean R. For example: R=10ms/pkt means that the sender S sends out packets with a mean time of 10ms/pkt. You must be able to specify the value of R on the command-line for the program running process S.

Define: **success** = (number of packet received / number sent) * 100%

Since the inter-packet time is a random variable, you will need to send sufficiently large number of packets to get an accurate measure of expected value of **success**. Also, reset the random seed and repeat the experiment multiple times to ensure that the measured success is accurate. This is a general strategy for experimentally measuring any random variable.

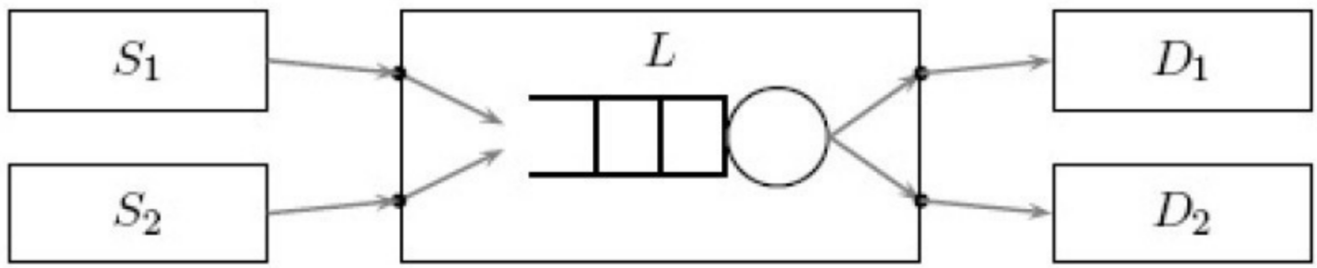
1.1 In addition to your code, generate a README describing how to run all three programs. Include at a minimum the following information:

1. **Names of group members**
2. **Emails of group members**
3. **Platform (e.g. Linux, Windows, Mac)**

4. **Compilation instructions**
5. **Run-time instructions (command-line arguments, etc.)**

1.2 Plot success (defined above) vs. R for different values of R, where 5 ms/pkt < R < 20 ms/pkt, and comment.

Step 2: Extend to 2 senders and 2 receivers



Modify the system so that it can support 2 flows: (S1,D1) and (S2,D2). Both going through the same router L with a single packet queue, that is, packets from both flows go through the same queue. The inter-packet time for both senders is a Poisson distribution as in **Step 1**. S2 sends with a mean wait time of **R** ms/pkt and S1 sends with mean wait time of **2R** ms/pkt. The service rate of the router L is still 10ms/pkt.

2.1 Plot success of both flows (S1,D1) and (S2,D2) vs R, where 5 ms/pkt < R < 20 ms/pkt, and comment.

2.2 From the above plots, observe and report at what value of R, does the success for flow (S1,D1) become 60% (approximately). We will use this in Step 3 of the project.

Step 3: Implement Priority

Now, let us assume that the flow (S1,D1) is real-time traffic and the inter-packet time is an exponential random variable with mean 2R (where R was obtained in Task 2.2) and flow (S2,D2) is traffic from a file transfer and the inter-packet time is an exponential random variable with mean R obtained in Task 2.2. To improve Quality of Service (QoS) from S1 to D1, prioritize the flow (S1,D1).

Implement the priority switching giving a higher priority to the flow (S1,D1)

Hint: One implementation of such a priority based routing would involve maintaining two separate queues in the router for both flows and servicing one queue with a higher priority than the other. Consider at the beginning equal split of the available buffer capacity over the two flows.

3.1 Compute and plot the success for (S1,D1) and (S2,D2) as function of R, where 5 ms/pkt < R <

20 ms/pkt for each of the flows. Compare the success for (S1,D1) and (S2,D2) with those observed in Task 2.1 and comment your observations.

3.2 Plot the average length of each queue vs. R. Then start changing the total buffer capacity. Plot and comment the impact of buffer capacity for low arrival rate, close to system capacity and under overload condition (meaning under relatively low, medium and high losses!) on average length of each queue. Comment.

3.3 Now, start changing the service rate of the router L. Assume the R = 12 and an equal split of the initial buffer size over the two queues. Plot loss rate and delay of (S2, D2) and average length of the queue for this flow vs. service rate of router, and comment. Determine the service rate such that you do not observe any losses in the queue.

3.4 Repeat the experiments from 3.3 changing the buffer length to half and double the original buffer size, assuming an equal split of the initial buffer size over the two queues. Comment. Focus on the impact of the buffer size as compared to the impact of router processing speed on loss rate and delay of flow (S2, D2).

POLICIES AND GUIDELINES

- Discussion on the interpretation of the assignment, the reasonable assumption, background information, references or programming language itself is highly encouraged across teams and is not regarded as a violation of Academic Honest and Integrity Policy.

- Fill in the gap yourself. You are free to choose anything not explicitly specified in this assignment with reasonable justification. Remember that there can be several ways to achieve the same goal.

- Timer control does not have to be precise. Keep it simple.

- You are flexible in all design choices not explicitly specified in this assignment. All such choices must be documented in the project report.

Submission Details

In total, each team must submit the following:

> **(1) All source code, including README file and compilation Makefile/scripts used. The code should include all programs used to run S, D, and L nodes. The router code should be able to handle either one or two flows. This should be in an archive (.zip, .tar, etc.) named:**
> `EE122project2-(last name)-(last name).(extension)`
>
> **(2) Writeup for experimental results. This report should include both students' names, the answers to all questions. This should be a PDF file named:**
> `EE122project2-report-(last name)-(last name).pdf`