Computer Networks Fall 2013: Project 2
Date Assigned: 10/14/2013
Date Due: 12/5/2013

1 Project Overview

This project consists of two parts. The first part is the simulation of a simple queueing system with a finite buffer to study the packet loss probability as a function of the buffer size and the traffic intensity. The second part is the simulation of the Ethernet exponential backoff algorithm. We will implement both project in Python using SimPy. You will find the required documentation at the following site `https://simpy.readthedocs.org/en/latest/contents.html#`. The site contains instructions to install SimPy and many examples. The following two sections (Sections 2 and Section 2) describe the two parts. The details of the deliverables (which are somewhat sketchy at this time) is described in Section 4.

2 Part 1

Figure 1 shows a simple queueing system model of output port of a router.

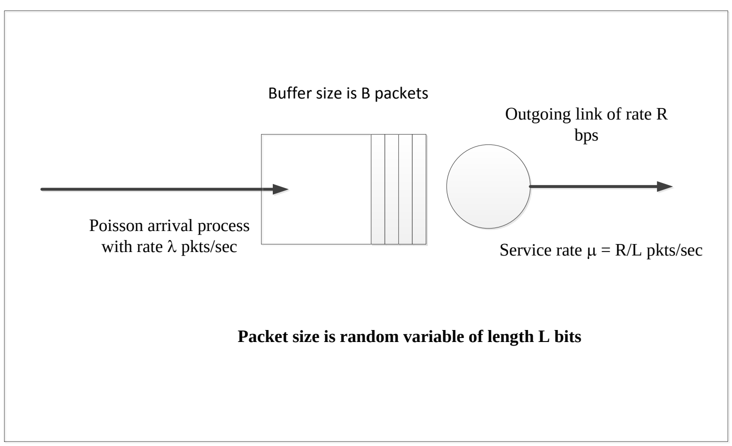

Figure 1: A finite buffer queue

Packets arrive following a Poisson process with rate λ. Packets are of variable length and transmitted on the link of rate R bps. We will assume that the effective service time of a packet is negative exponentially distributed with rate parameter μ packets per second. To keep things simple we will assume that $\mu = 1$ pkts/sec. The buffer size if B packets.

You are required to write a Python code using SimPy to simulate the above system and plot (or tabulate) the following results

1. For $\lambda = 0.2, 0.4, 0.6, 0.8, 0.9, 0.99$ and $B = 10, 50$ determine the packet loss probability P_d.
2. Compare the above results using the theoretical formula that you derived in Problem 5 in Assignment 4.

3 Part 2

In this part you will simulate and analyze the binary exponential backoff algorithm of the IEEE 802.3 Ethernet protocol. Before you get started you should read Section 5.3.2 (Random Access Protocols) of the text. We will cover it in class but you should read ahead.

Figure 2 shows the simulation model.

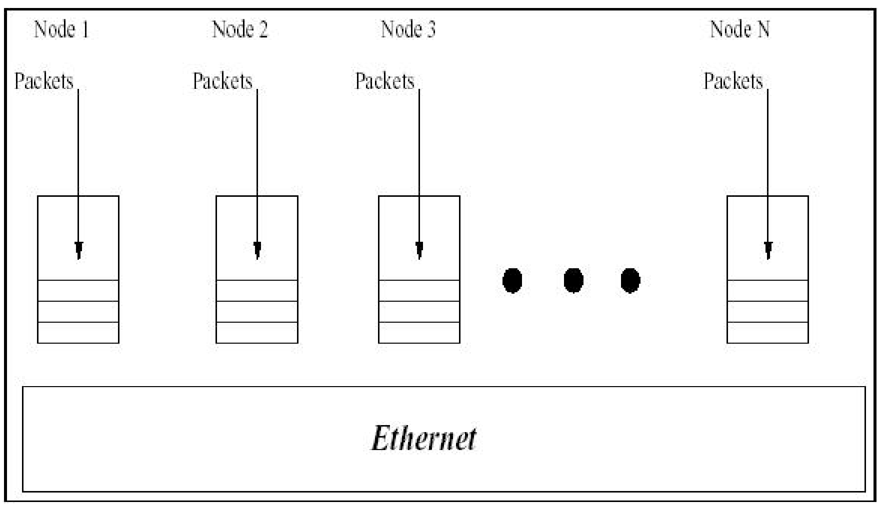

Figure 2: Ethernet simulation model.

In order to develop the simulation model, we will make the following assumptions:

1. We will assume that time is slotted into equal length of time slots. In the subsequent discussion, the length of the time slot will be denoted by T_s.
2. We will let N denote the number of hosts. As before, λ will denote the mean arrival rate of packets. We will assume the hosts are identical, and each hosts receiving packets with a mean arrival rate of λ packets/second. The arrival process follows a Poisson process with rate parameter λ pkts/sec (same as Part 1 of the project).
3. Hosts can transmit only at slot boundaries.

4. If at a particular slot boundary there are more than one host ready to transmit, there will be a collision. When hosts collide, they will schedule their retransmission using the following binary exponential backoff algorithm. The number of slots to delay after the n^{th} retransmission attempt is chosen as a uniformly distributed integer in the range $0 < r \leq 2^K$, where $K = \min(n, 10)$.

5. In this phase, we will be interested in plotting the throughput where throughput is defined as the number of successful transmission per time unit. In the simulation, you can count the number of slots in which there is successful transmission and divided that by the total number of slots that you simulate.

Develop a simulation model the above system. You have to get the following results:

1. Plot or tabulate the throughput as a function of λ with the binary exponential backoff algorithm as described above. Slot time $T_s = 1$ and number of hosts $N = 10$. Obtain the throughput for the values of $\lambda = 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09$.

2. Do the same as above with the following modification to the binary exponential backoff algorithm: the number of slots to delay after the n^{th} retransmission attempt is chosen as a uniformly distributed integer in the range range $0 < r \leq K$, where $K = \min(n, 1024)$. This is linear backoff. Again obtain the throughput for the values of $\lambda = 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09$.

A Sample Execution

In order to understand how the backoff algorithm works it will help you to work though this problem. We consider 3 hosts operating following the same slotted system described above. Each host maintains 3 variables

L: The number of packets in the queue.

N: The number of times the packet at the head of the queue has been *retransmitted*. When a new packet comes to the head of the queue n is reset to 0.

S: The slot number when the next transmission attempt will be made for the packet at the head of the queue.

Figure 3 shows the state of the variables for the three nodes.

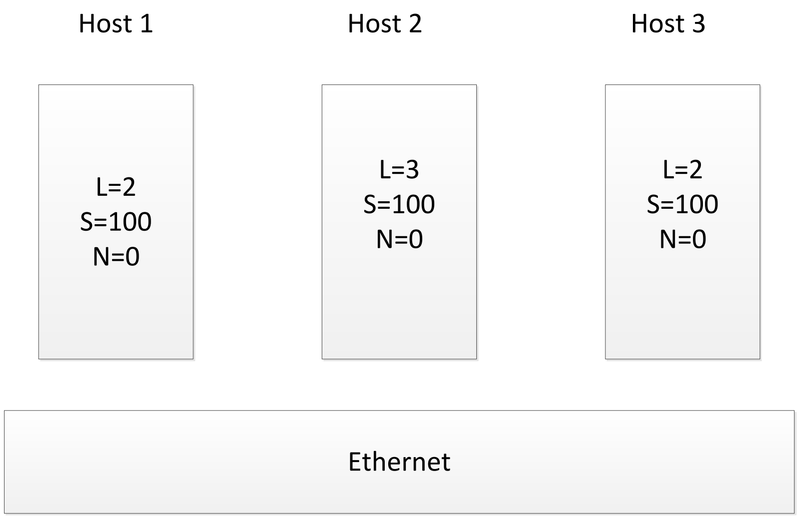

Figure 3: 3 Node Ethernet simulation model

Assuming that the random are drawn from the following sequence, determine number of slots requires to transmit all the packets from the three nodes. Random number sequence:

```
0.6621015449 0.0598288625 0.5979568986 0.0940302045 0.9731501949
0.0662842158 0.5671123588 0.2960073405 0.0836140907 0.1982544372
0.3973937067 0.1825762354 0.7300244789 0.9750743033 0.4226574674
0.7283845034 0.9393420117 0.3463454130 0.0346137777 0.5296165026
0.7221479421 0.4893986084 0.7332225929 0.3619475896 0.3159643647
0.7164982632 0.8700478233 0.4864938182 0.2679576606 0.1540246727
0.5281929232 0.6834302917 0.4536605629 0.7499270914 0.0581091905
0.0344902470 0.6920460807 0.2915925588 0.8664808415 0.7130973549
0.1030703052 0.6871530802 0.7494474363 0.6699161683 0.2188189046
0.5913891383 0.9112765496 0.0002334074 0.5658022752 0.0542878532
```

4 Deliverables and Other Details

This section will be flushed out in the next few days. Here are some preliminary information:

1. Group size: 2
2. Coding language: Python
3. Submission date: 12/5/2013