1 Introduction

In this phase, we will extend the single server and single queue to multiple independent hosts that communicate using the Ethernet protocol. The goal is to analyze the behavior of the Ethernet protocol, namely the **throughput of the channel**. The model of the system is shown in Figure 1.

![Figure 1: Model of the Ethernet.](image)

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, $\lambda$ will denote the mean arrival rate of packets. We will assume the hosts are identical, and each host receives packets with a mean
arrival rate of \( \lambda \) packets/second. The arrival process follows the exponential distribution (same as Phase I 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 transmissions per time unit. In the simulation, you can count the number of slots in which there is successful transmission and divide that by the total number of slots.

Rework the code you have developed in Phase I to model the above system. Assume that \( \mu = 1 \text{ packet/second}. \)

2 Submission Guidelines

You have to get the following results:

- **Exponential Backoff:** plot the throughput as a function of \( \lambda \) 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 \( \lambda = 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09 \). Provide a table of results of throughput with the corresponding lambda.

- **Linear Backoff:** 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 \( 0 < r \leq K \), where \( K = \min\{n, 1024\} \). In this case, the range is reduced. Provide a table of results of throughput with the corresponding lambda.

- **Random backoff:** plot the throughput as a function of \( \lambda \) using the Slotted Aloha backoff algorithm: the number of slots to delay after the \( n^{th} \) retransmission attempt is chosen as a uniformly distributed integer in the range: \( 0 < r < M \), where \( M = 5 \) (max number of slots to delay). Obtain the throughput for values of \( \lambda = 0.001, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.09 \). Provide a table of results of throughput with the corresponding lambda.

- **Writeup:** give a 1 to 2 page write up on the results. Explain why you get such results and provide comparisons of the 3 backoff algorithms.

- **Hardcopy of Source Code:** provide a hardcopy of the source code as in Phase I (i.e. if you use anything from the library, you don’t have to submit those. Also try to print 2 pages on 1 sheet of paper.)

- **Softcopy of Source Code:** you need to submit a softcopy of your code through the **handin** utility (user: oliveira, dest_folder: project2) You need to submit your source files, a README file describing how to run your program (input parameters) and a **Makefile** so
that we are able to compile and run your program in **LINUX**. For more information on the handin utility: