Keshav Tirumurti 999290785
Alex Soong 998857734

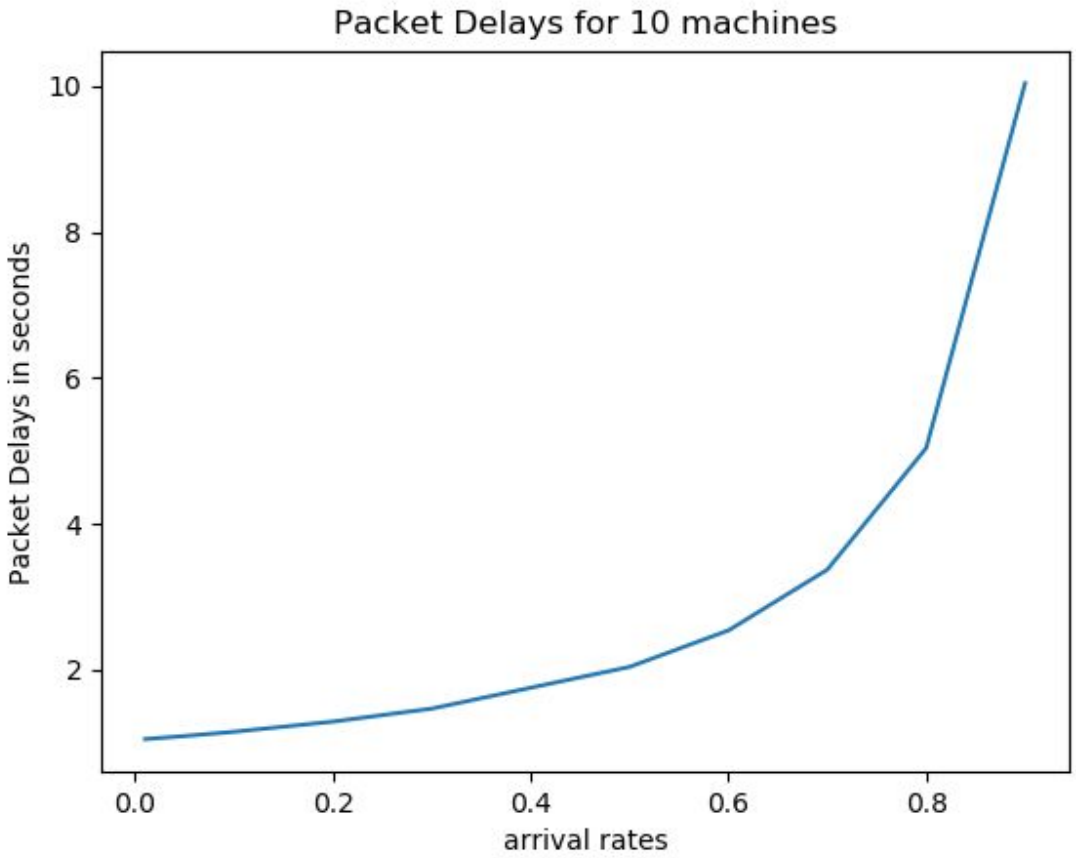
Implementation

In our Token Passing Protocol implementation, we created an array of hosts (each host with its respective buffer) and a GEL to keep track of all the events in the LAN. We define two types of events: a packet and a token. Upon initialization, we populate the GEL with random events and designate the current token in the system to a random host in the array. In order to process these events, we create a nested loop: the outer loop traverses through the events in the GEL, and the inner loop traverses through the hosts. When the current GEL event is a packet, we look at the packet's src host and append the event to the corresponding host's buffer in the array. If the current GEL event is a token, we traverse through the inner loop through all the hosts. In this traversal, we find the host that currently has the token. Next, we copy that host's buffer into a frame, and then empty the contents of the buffer. Next, we pass the token onto the next chronological host in the array. We repeat these steps for every event in the GEL.

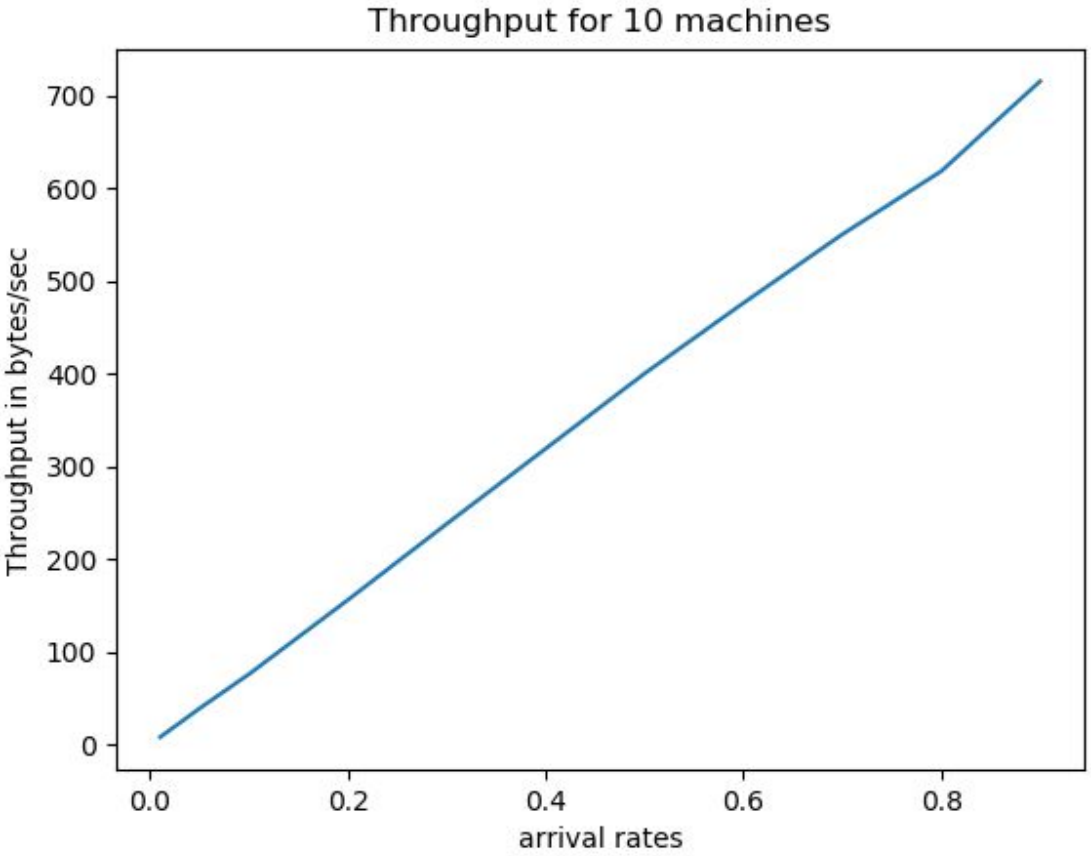
Source Code:

Our source code is attached as the file LAN.py.

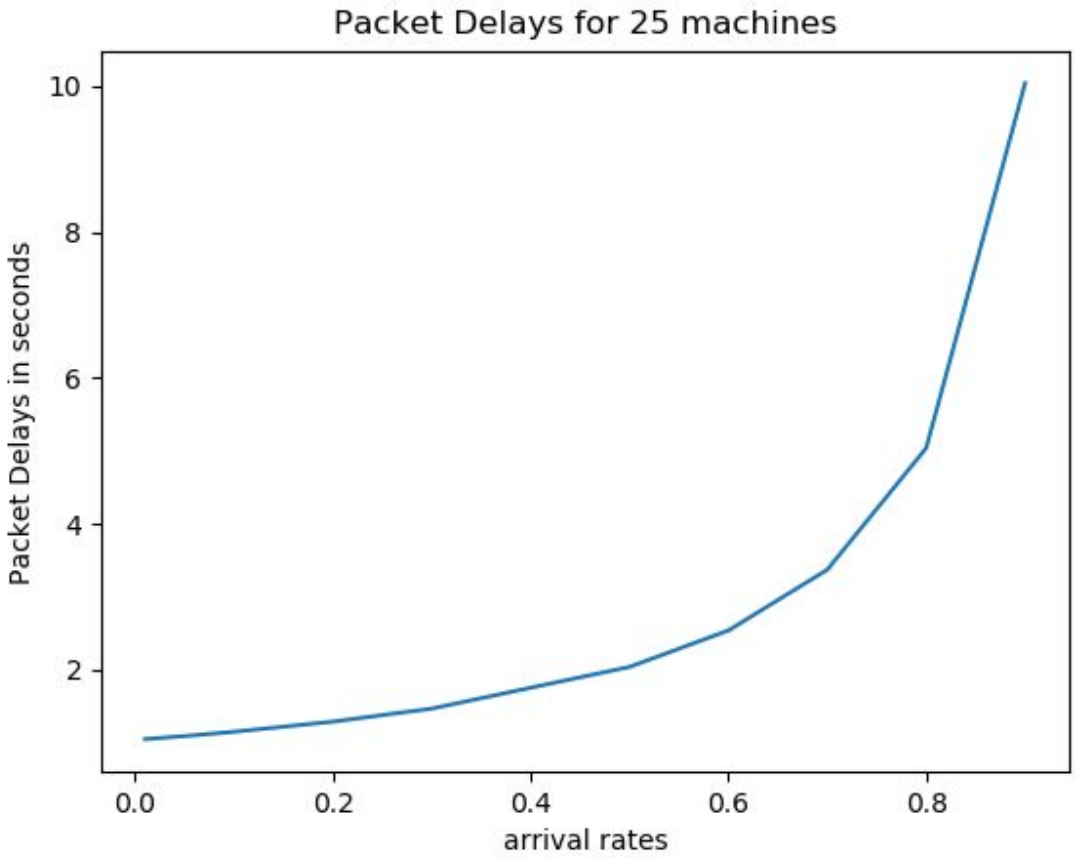
Results:



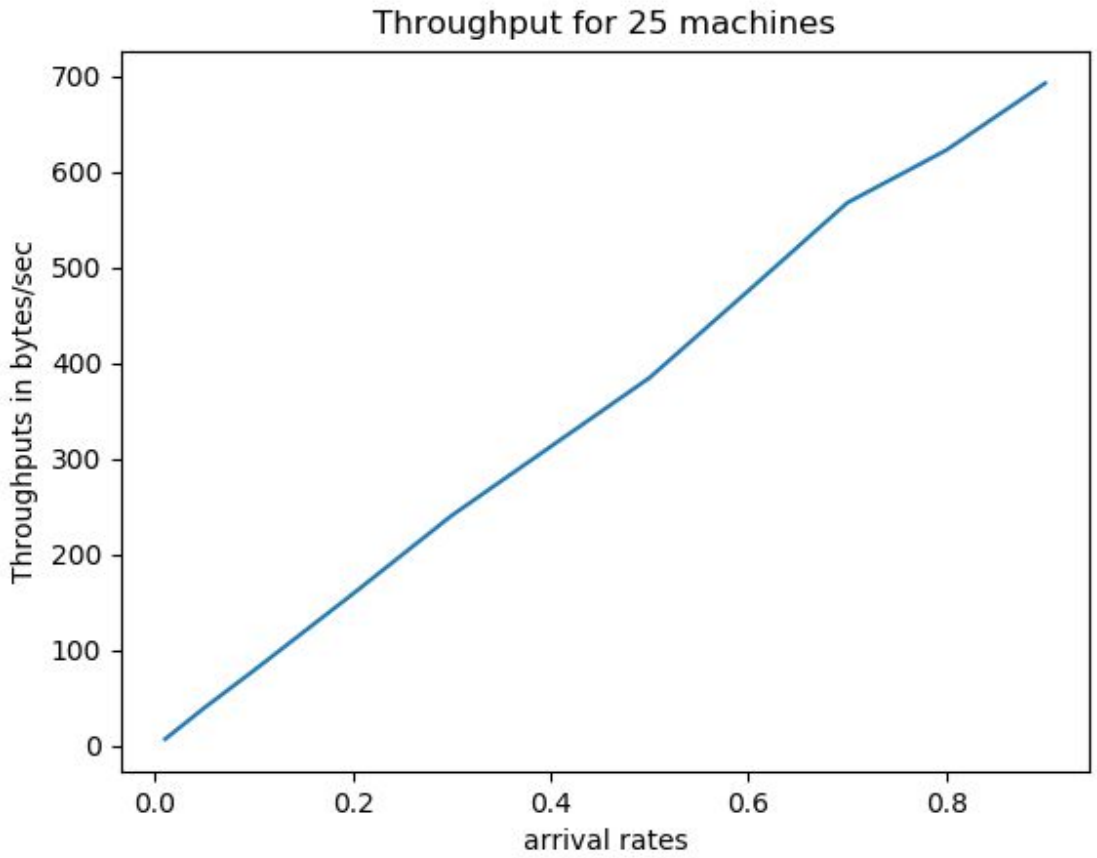
Keshav Tirumurti 999290785
Alex Soong 998857734



Keshav Tirumurti 999290785
Alex Soong 998857734



Keshav Tirumurti 999290785
Alex Soong 998857734



Analysis of the Results

Throughput for both N=10 and N=25 grew linearly as arrival rate increased. The packet delays for the two host sizes, however, grew exponentially as arrival rate increased. According to our results, the delays and throughputs were very similar for both sizes of N. This suggests that both the throughput and the average packet delays are dependent on arrival rate, and do not depend on the size of N.