Lab 2

Kashturi Vasanthan
Parvathi Krishnan

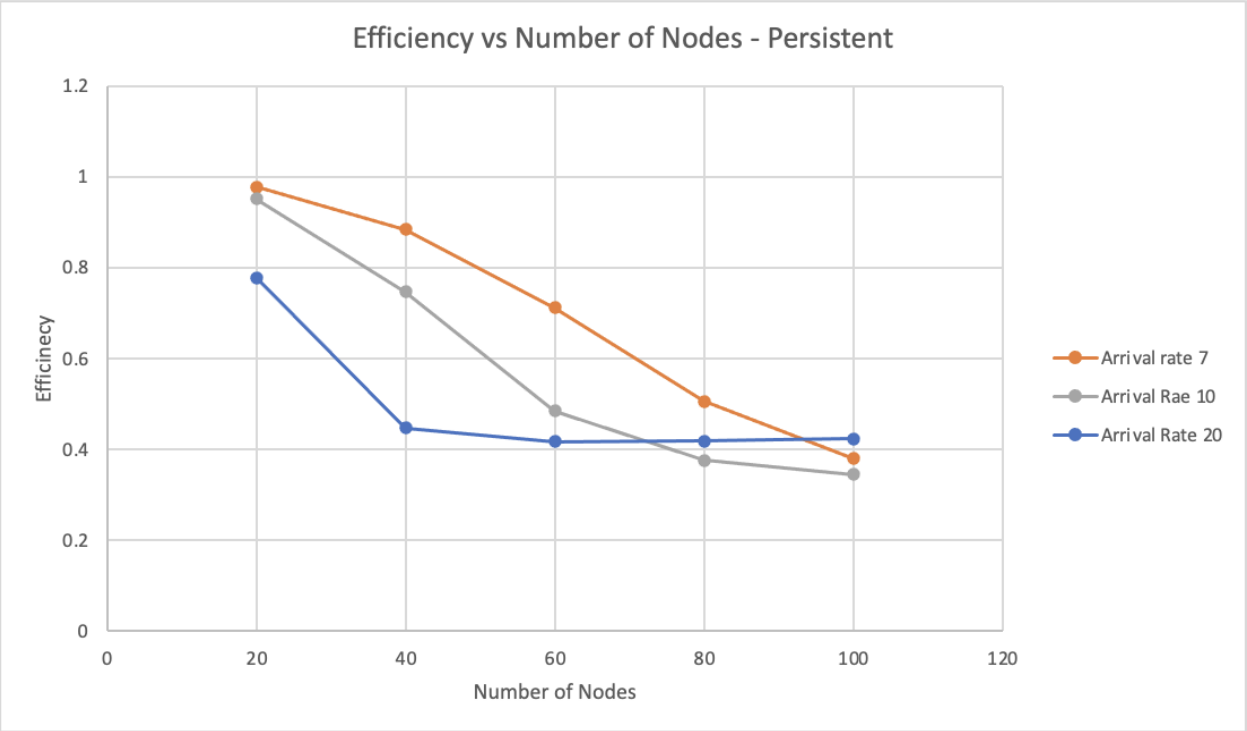
Table of Contents

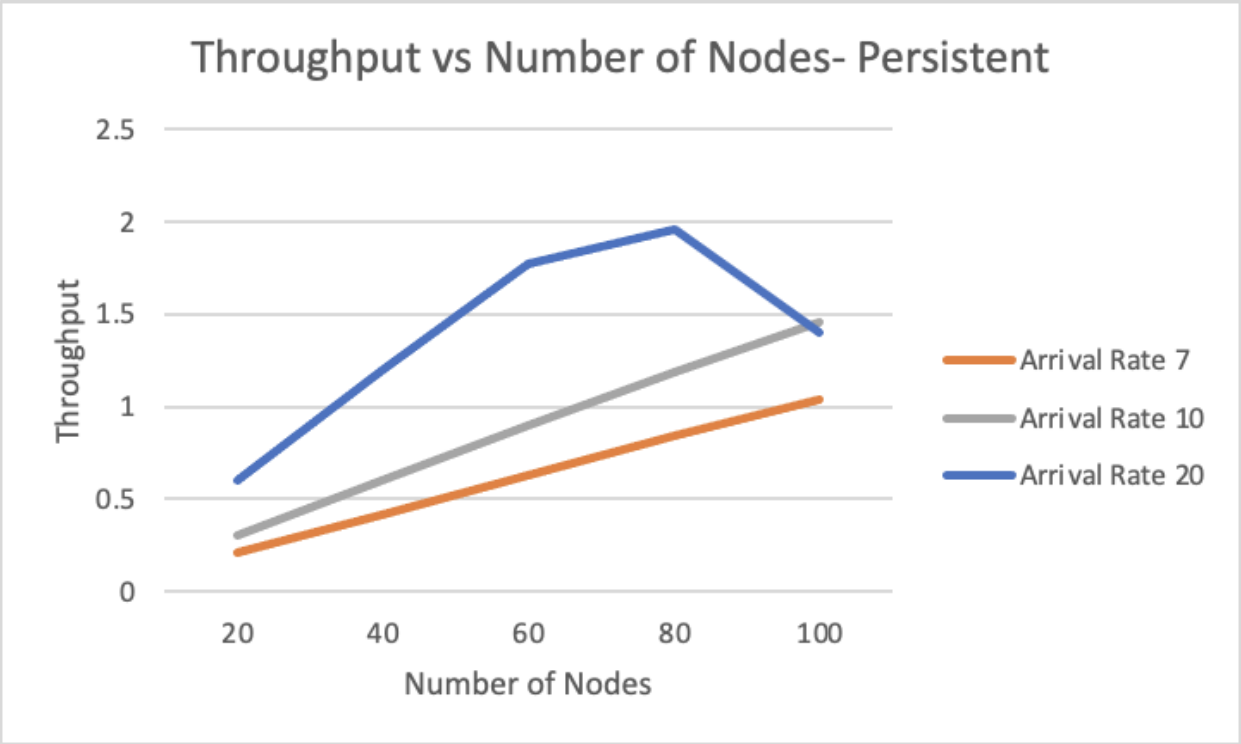
Lab 2

Question 1

1. Simulate a persistent CSMA/CD protocol.

2. Show the efficiency and throughput (in Mbps) of the LAN as a function of N (20, 40, 60, 80, and 100) for A = 7, 10 and 20 Packet/sec, R= 1 Mbps, and L = 1500 bits.





3. Comment on the behavior of the graphs.

   As you can see, as the number of nodes increases, the efficiency of our simulation decreases. Similarly, as the arrival rate increased, the efficiency of our simulation decreased as well.

4. In particular, define your variables.

```
public class Simulator {
   double D = 10;
   double S = (2 / 3);
   double R = 1e6;
   int L = 1500;
   double simT;
   int N;
   int A;
   double Ttrans;
   double Tprop;
   boolean isPersistent;
   Node[] nodes;
}
```

```
public class Node {
   double time;
   int numOfCollisions;
```

```
    int busyBuffer;
    Queue<Double> queue;
}
```

```
public class Result {
    int N;
    int arrivalRate;
    double efficiency;
    double throughput;
}
```

5. Should there be a need, draw diagrams to show your program structure.
   Pseudocode

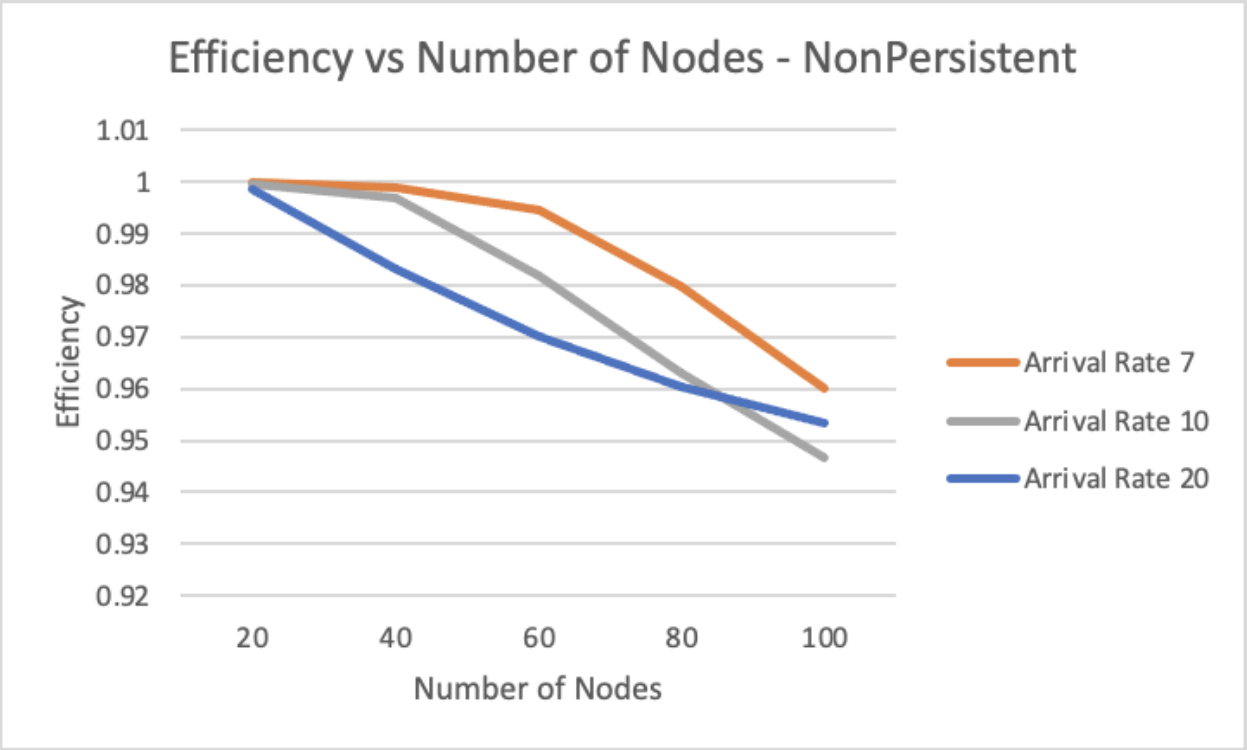
```
// Variables used to keep track of transmission attempts and number of
successful
        // transmissions



        // Generate the nodes with a queue of frames/packets using a poisson
distribution
        // all within the range of the simulation

        // Running the actual simulation

            // Select which node should transmit next, this node should have the
frame with
            // the lowest timestamp

            // If no nodes are able to transmit, the simulation is finished


            // Check if any other nodes aside from the sending node(the one that
            // transmitted) experiences a collision

                // The time it takes for the first bit to be received by the
current node from
                // the sender

                // The time it takes for the last bit to be received by the
current node from
                // the sender
```

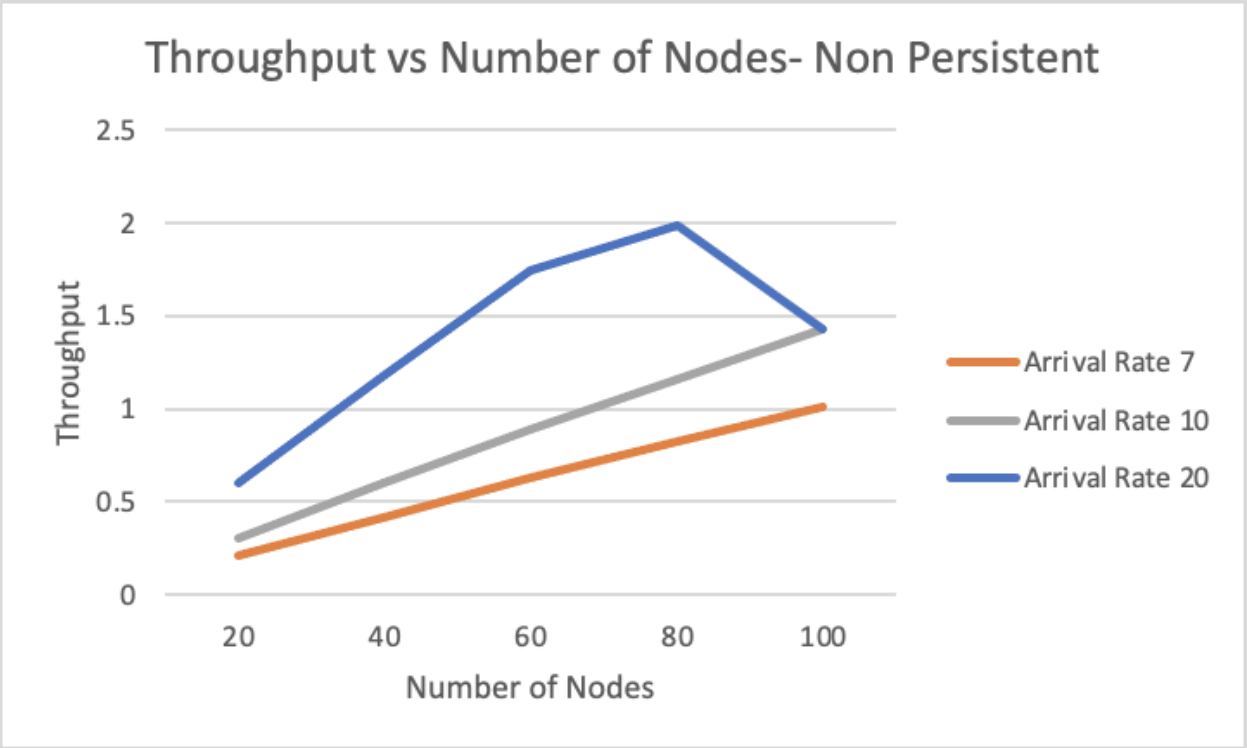
```
               // Compare the current node's sense time to the transmitted
frame's
               // arrival time at the current to check if there is a collision
or not

                   // If the node's collision counter is greater than Kmax
which is 10, then we
                   // drop the frame from the queue
                   // and reset the collision counter and reset our sense time

                       // Otherwise, we back off this node to the sending node
for a certain amount of
                       // time

           // On the current sending node, we check if there were any
collisions

               // Any node that sense between the the start and end times, the
node will find
               // the bus busy, hence,
               // we readjust the sense time

                       // in the case of the simulation being persistent we
schedule our next bus-sense
                       // right at time end and immediately try again

                       // in the non persistent case, we add an exponential
backoff to the current
                       // sensing time

               // Reset the collision counter, remove the frame from the queue,
increment our
               // success and reset our sense time

           // In the case of a collision we back off the sending node by a
given amount
               // if the number of collisions exceeds Kmax, we drop the frame
and reset the
               // sense time for the sending node
               // Otherwise we just back off exponentially

       // Return our appropriate results back
```

6. Explain how you compute the performance metrics.
   Throughout the simulation, we have a variable that keeps account of all the attempts and the number of packets that were unvisited. After the simulation is done efficiency is calculated by `(success * 1.0 / (attempts))` and throughput is calculated by `Math.abs((success * L / simT)/1000000))/`

Question 2

Show the efficiency and throughput (in Mbps) of non-persistent CSMA/CD protocol for the same network parameters used in question
1. Comment on the graph and compare between the results obtained in question 1 and question
Compared to the graph in 1, the efficiency behaved in the same manner, and the throughput seemed to behave in a very similar manner as well.





7. In particular, define your variables.

```
public class Simulator {
   double D = 10;
   double S = (2 / 3);
   double R = 1e6;
   int L = 1500;
   double simT;
   int N;
   int A;
   double Ttrans;
   double Tprop;
   boolean isPersistent;
   Node[] nodes;
}
```

```
public class Node {
   double time;
   int numOfCollisions;
   int busyBuffer;
   Queue<Double> queue;
}
```

```
public class Result {
```

```
    int N;
    int arrivalRate;
    double efficiency;
    double throughput;
}
```

8. Should there be a need, draw diagrams to show your program structure.
   Pseudocode

```
// Variables used to keep track of transmission attempts and number of
successful
       // transmissions



       // Generate the nodes with a queue of frames/packets using a poisson
distribution
       // all within the range of the simulation

       // Running the actual simulation

           // Select which node should transmit next, this node should have the
frame with
           // the lowest timestamp

           // If no nodes are able to transmit, the simulation is finished



           // Check if any other nodes aside from the sending node(the one that
           // transmitted) experiences a collision

               // The time it takes for the first bit to be received by the
current node from
               // the sender

               // The time it takes for the last bit to be received by the
current node from
               // the sender

               // Compare the current node's sense time to the transmitted
frame's
               // arrival time at the current to check if there is a collision
or not
```

```
                    // If the node's collision counter is greater than Kmax
which is 10, then we
                    // drop the frame from the queue
                    // and reset the collision counter and reset our sense time

                        // Otherwise, we back off this node to the sending node
for a certain amount of
                        // time

            // On the current sending node, we check if there were any
collisions

                // Any node that sense between the the start and end times, the
node will find
                // the bus busy, hence,
                // we readjust the sense time

                        // in the case of the simulation being persistent we
schedule our next bus-sense
                        // right at time end and immediately try again

                        // in the non persistent case, we add an exponential
backoff to the current
                        // sensing time

                // Reset the collision counter, remove the frame from the queue,
increment our
                // success and reset our sense time

            // In the case of a collision we back off the sending node by a
given amount
                // if the number of collisions exceeds Kmax, we drop the frame
and reset the
                // sense time for the sending node
                // Otherwise we just back off exponentially

        // Return our appropriate results back
```

9. Explain how you compute the performance metrics.
   Throughout the simulation, we have a variable that keeps account of all the attempts and the number of packets that were unvisited. After the simulation is done efficiency is

calculated by `(success * 1.0 / (attempts))` and throughput is calculated by `Math.abs((success * L / simT)/1000000))/`