DEPARTMENT OF ELECTRONICS & COMPUTER SCIENCES

# Final Project Report
Project #2

Professor: Dr. Chul-Ho Lee
COMPUTER NETWORKS - ECE-5534

Sannithi vinod kumar (903636939)
Rosalin dash (9036944640)

# INDEX

## Table of Contents

# INTRODUCTION

**Carrier sense multiple access/collision detection** (**CSMA/CD**) is a MAC (media access control) protocol. It defines how network devices respond when two devices attempt to use a data channel simultaneously and encounter a data collision.

The CSMA CD have some rules for the device after the collision. The medium is sometimes used by multiple data nodes, so each data node receives transmissions from each of the other nodes on the platform / medium.

There are several CSMA access modes:

1. 1 - persistent
2. P - persistent
3. O - persistent

## 1 - PERSISTENT

1-persistent is used in CSMA/CD systems, like Ethernet. This mode waits for the medium to be idle, then transmits data.

## P - PERSISTENT

P-persistent is used in **CSMA/CA** systems, like Wi-Fi. This mode waits for the medium to be idle, then transmits data with a probability p. If the data node does not transmit the data (a probability of 1-p), the sender waits for the medium to be idle again and transmit the data with the same probability p this continues until the mode transmits with the probability p.

## O - PERSISTENT

Each node is assigned an order by the supervisory node, this mode assigns a transmission order to each data node. When the medium becomes idle, the data node next in line can transmit data. The data node next in line waits for the medium to be idle again and then transmits its data. After each data node transmits data, the transmission order is updated to reflect what data nodes have already transmitted, moving each data node through the queue

# GOAL

The main objective of this course project is to reinforce your understanding on medium access control (MAC) protocols and let you have more visible experience through the design and implementation of a custom-made network simulator with focus on the performance evaluation of MAC protocols. The MAC protocol of interest in this project is 1-persistent CSMA/CD with Binary Exponential Back off. This project is done based on the off-the-shelf network simulators such as ns - 2, ns - 3, OPNET, and OMNeT++. While I can still refer to the existing simulators, ultimately you have to implement your own discrete-event network simulator from scratch. This project requires a fair amount of work, but this is where you draw upon (and also build up) your knowledge and experience. This document is intended to ease the process of figuring out what to do in this project.

# PREREQUISITE FOR CODE RUN

- JAVA installed in your system preferable Java version10 and above.
- Needs User input from terminal as per the program request to get necessary information's as per the Goals

# DESCRIPTION OF CODE

The code is based on the Real Time Operations. The entire code is simulated on a given timeslot which is user inputted data. The loops run over the timeslot with a constant increment and gives each node equal opportunity to generate packet as per the given probability.

The random variable is generated by using Math.random() predefined java operation. On the basis of random value probability of packet generation is decided. This random function is also used to decide the destination IP address to which the packet is forwarded.

Once the packet is generated, we check the nodes which has packets to be transmitted. If only one node has packet, its back off time is zero and channel is idle to transmit we then remove the packet from buffer and place it on the transmission channel to transmit it to the destination IP. During this transmission we keep the channel busy for next 10 iterations in the meanwhile nodes still generates packets with the same probability. Once the channel falls back to idle after 10 iterations other nodes sense the channel and the node whose back off time is zero gets the chance to transmit its packet.

Considering the channel is idle and two nodes gets to the mode of packet ready we consider it as collision and update the necessary back off time with using the binary exponential algorithm, where after the $m^{th}$ collision the station waits for a random number of time slots, which is chosen uniformly at random from the set { 0,1,2,…….$2^m$ } where m = min { 10, m}, before attempting to retransmit. After 16 collisions the station gives up.

Considering the above procedure, we update various statistics values like

- The number of frames generated
- Average number of frames generated
- The total number of frames lost due to buffer overflow
- The total number of frames waiting in the buffer
- The number of frames Successful transmission
- Throughput
- Channel utilization
- Channel waste
- Delay

# OBSERVATION

Observation are taken into account with the below Input data sets as mentioned in the Samples. Mentioned Samples are collected keeping

- Back OFF Minimum value as 10
- Channel Busy for consecutive 9 time slots if a channel is transmitting
- Packet retransmission count of 15 after which the packet will be dropped

1. Sample Test 1

   Input Data Set

```
Enter the number of Nodes
6
Enter the buffer size for each Nodes
10
Enter the number of TimeSlot
1000
Enter the probability value for generating Packet
0.1
```

Output Data Set

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|                  PER NODE DETAIL INFORMATIONS                    |
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|Average Number of frame generated by 10.0.0.1          | 1.0      |
+------------------------------------------------------------------+
|Average Number of frame generated by 10.0.0.2          | 1.0      |
+------------------------------------------------------------------+
|Number of frame generated by 10.0.0.5                  | 1000     |
+------------------------------------------------------------------+
|Average Number of frame generated by 10.0.0.5          | 1.0      |
+------------------------------------------------------------------+
|Number of frame generated by 10.0.0.4                  | 1000     |
+------------------------------------------------------------------+
|Average Number of frame generated by 10.0.0.6          | 1.0      |
+------------------------------------------------------------------+
|Average Number of frame generated by 10.0.0.3          | 1.0      |
+------------------------------------------------------------------+
|Average Number of frame generated by 10.0.0.4          | 1.0      |
+------------------------------------------------------------------+
|Number of frame generated by 10.0.0.6                  | 1000     |
+------------------------------------------------------------------+
|Number of frame generated by 10.0.0.1                  | 1000     |
+------------------------------------------------------------------+
|Number of frame generated by 10.0.0.3                  | 1000     |
+------------------------------------------------------------------+
|Number of frame generated by 10.0.0.2                  | 1000     |
+------------------------------------------------------------------+
```

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|                   OVERALL SIMULATION RESULTS                     |
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|Channel Waste                                          | 0.399    |
+------------------------------------------------------------------+
|Successful Transmission Frame Count                    | 61       |
+------------------------------------------------------------------+
|Throughput                                             | 0.061    |
+------------------------------------------------------------------+
|Total Number of Frames                                 | 6000     |
+------------------------------------------------------------------+
|Total Number of Frames Lost                            | 5874     |
+------------------------------------------------------------------+
|Average of Total Frames generated                      | 6.0      |
+------------------------------------------------------------------+
|Awaiting Frames                                        | 65       |
+------------------------------------------------------------------+
|Average Delay                                          | 2.773    |
+------------------------------------------------------------------+
|Channel Utilization                                    | 60.1%    |
+------------------------------------------------------------------+
```

2. Sample Test 2

Input Data Set

```
Enter the number of Nodes
10
Enter the buffer size for each Nodes
15
Enter the number of TimeSlot
10000
Enter the probability value for generating Packet
0.3
```

Output Data Set

| PER NODE DETAIL INFORMATIONS | |
|---|---|
| Average Number of frame generated by 10.0.0.10 | 1.0 |
| Average Number of frame generated by 10.0.0.1 | 1.0 |
| Number of frame generated by 10.0.0.10 | 10000 |
| Average Number of frame generated by 10.0.0.2 | 1.0 |
| Number of frame generated by 10.0.0.5 | 10000 |
| Average Number of frame generated by 10.0.0.5 | 1.0 |
| Number of frame generated by 10.0.0.4 | 10000 |
| Average Number of frame generated by 10.0.0.6 | 1.0 |
| Average Number of frame generated by 10.0.0.3 | 1.0 |
| Number of frame generated by 10.0.0.7 | 10000 |
| Average Number of frame generated by 10.0.0.4 | 1.0 |
| Number of frame generated by 10.0.0.6 | 10000 |
| Number of frame generated by 10.0.0.1 | 10000 |
| Average Number of frame generated by 10.0.0.9 | 1.0 |
| Number of frame generated by 10.0.0.3 | 10000 |
| Average Number of frame generated by 10.0.0.7 | 1.0 |
| Number of frame generated by 10.0.0.2 | 10000 |
| Average Number of frame generated by 10.0.0.8 | 1.0 |
| Number of frame generated by 10.0.0.9 | 10000 |
| Number of frame generated by 10.0.0.8 | 10000 |

| OVERALL SIMULATION RESULTS | |
|---|---|
| Successful Transmission Frame Count | 486 |
| Throughput | 0.0486 |
| Total Number of Frames Lost | 99354 |
| Average of Total Frames generated | 10.0 |
| Channel Utilization | 48.6% |
| Channel Waste | 0.514 |
| Total Number of Frames | 100000 |
| Awaiting Frames | 160 |
| Average Delay | 2.7226 |

## 3. Sample Test

### Input Data Set

```
Enter the number of Nodes
8
Enter the buffer size for each Nodes
20
Enter the number of TimeSlot
100000
Enter the probability value for generating Packet
0.5
```

### Output Data

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|                        PER NODE DETAIL INFORMATIONS                 |
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|Average Number of frame generated by 10.0.0.1              | 1.0     |
+---------------------------------------------------------------------+
|Average Number of frame generated by 10.0.0.2              | 1.0     |
+---------------------------------------------------------------------+
|Number of frame generated by 10.0.0.5                      | 100000  |
+---------------------------------------------------------------------+
|Average Number of frame generated by 10.0.0.5              | 1.0     |
+---------------------------------------------------------------------+
|Number of frame generated by 10.0.0.4                      | 100000  |
+---------------------------------------------------------------------+
|Average Number of frame generated by 10.0.0.6              | 1.0     |
+---------------------------------------------------------------------+
|Average Number of frame generated by 10.0.0.3              | 1.0     |
+---------------------------------------------------------------------+
|Number of frame generated by 10.0.0.7                      | 100000  |
+---------------------------------------------------------------------+
|Average Number of frame generated by 10.0.0.4              | 1.0     |
+---------------------------------------------------------------------+
|Number of frame generated by 10.0.0.6                      | 100000  |
+---------------------------------------------------------------------+
|Number of frame generated by 10.0.0.1                      | 100000  |
+---------------------------------------------------------------------+
|Number of frame generated by 10.0.0.3                      | 100000  |
+---------------------------------------------------------------------+
|Average Number of frame generated by 10.0.0.7              | 1.0     |
+---------------------------------------------------------------------+
|Number of frame generated by 10.0.0.2                      | 100000  |
+---------------------------------------------------------------------+
|Average Number of frame generated by 10.0.0.8              | 1.0     |
+---------------------------------------------------------------------+
|Number of frame generated by 10.0.0.8                      | 100000  |
+---------------------------------------------------------------------+
```

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|                  OVERALL SIMULATION RESULTS                      |
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|Successful Transmission Frame Count                     | 5416    |
+------------------------------------------------------------------+
|Throughput                                              | 0.05416 |
+------------------------------------------------------------------+
|Total Number of Frames Lost                             | 794416  |
+------------------------------------------------------------------+
|Average of Total Frames generated                       | 8.0     |
+------------------------------------------------------------------+
|Channel Utilization                                     | 54.16%  |
+------------------------------------------------------------------+
|Channel Waste                                           | 0.4584  |
+------------------------------------------------------------------+
|Total Number of Frames                                  | 800000  |
+------------------------------------------------------------------+
|Awaiting Frames                                         | 168     |
+------------------------------------------------------------------+
|Average Delay                                           | 0.23882 |
+------------------------------------------------------------------+
```

# CONCLUSION

The above simulation gave us a clear understanding about CSMA/CD and how the performance of CSMA/CD is affected by varying the number of nodes, time slot, buffer size and the probability of generating packet. By looking at the above statistics we can conclude that CSMA/CD has a throughput of approx. 0.05.

# References

[1]. **"Computer Networks"**, 5th edition, **by Andrew S. Tanenbaum and. David Wetherall**, Pearson, 2011.

[ 2] **"Computer Networking: A Top-Down Approach"**, 7th edition, **by James F. Kurose and Keith W. Ross**, Pearson, 2017.