Project 1

ECE 578

Thao Vo
Patrick Martin

Fall 2017

Project Description:

In this project, our group will look into the Distributed Coordination Function (DCF) of the 802.11 protocol in multiple different network topologies to evaluate the correlation between performance metrics such as throughput, number of collisions, and fairness index.

We are given two different scenarios for the network topologies: the concurrent communication (Figure 1 (a)) where Nodes A, B, C, D all reside within the same collision domain and all transmissions are received. This scenario is known as the exposed terminal. The second scenario is called the hidden terminal (Figure 1 (b)), where station A, B, C belong to two different collision domains and transmissions can be lost. When communication takes place between nodes, the traffic will be generated according to an exponentially distributed Poisson arrival process.

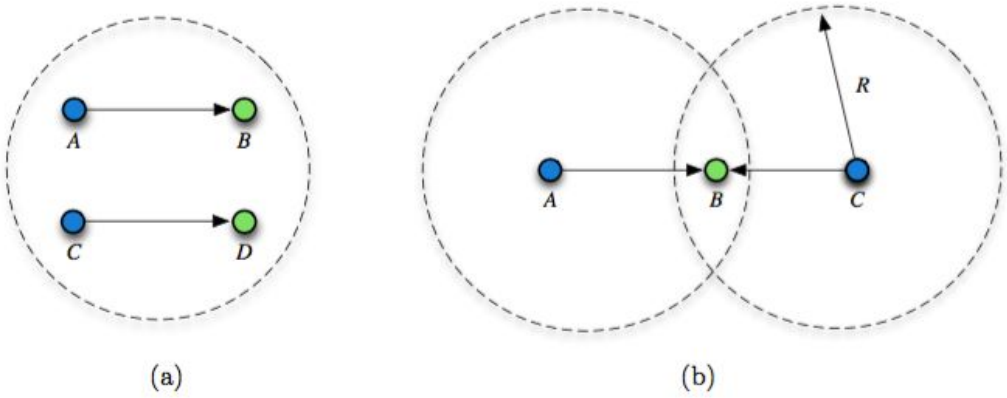

Figure 1: (a) Concurrent Communication/Exposed Terminal and (b) Hidden Terminal

Our group will model the behavior of a Carrier Sense Multiple Access(CSMA)/Collision Avoidance (CA) and CSMA/CA with Virtual Carrier Sensing (VCS) to simulate and evaluate the performance metrics between them. The main differences between CSMA/CA vs. VCS are as follows:

CSMA:

- Detect a collision early and abort the transmission.
- Each node tries to retransmit at a contention slot with probability p.
- If a collision occurs, it is detected at the end of the contention slot.
- If the transmission is a success, no other host tries to transmit till the end of the packet slot.

VCS:

- Collisions are receiver dependent, both transmission will be successful.

- Senses channel for DIFs (DCF Interframe Space), if the channel is busy continue to sense, otherwise select a random backoff time *b* in [0, CW] and reduce by one with every idle slot.
- If the channel becomes busy, freeze the counter. If counter = 0, send RTS with NAV (Network Allocation Vector). Receiver acknowledges via CTS after SIFS (Short Interframe Space).
- CTS reserves channel for sender, notifying possibly hidden stations; any station hearing the CTS should be silent for the NAV. Sender can now send data immediately.

Responsibilities:

While working on this project equally, each member had a different focus. Our group worked together to figure out each other's responsibilities, planned how to approach the work, and determined how to simulate and design our network environment for comparison, debugging of codes, and checking on all components of the final report.

Thao - Wrote the introduction, logic and implementation of codes, main focus in concurrent communication (1a CSMA/VCS), planning, and code reviews.

Patrick - Wrote the conclusion, main focus in hidden terminal (1b CSMA/VCS), graphed the charts, and code reviews.

Parameters & Values:

Table 1 shows the given values provided by the project documentation:

Parameter	Value	Parameter	Value
Data frame size	1,500 bytes	ACK, RTS, CTS size	30 bytes
Slot duration	20 μs	DIFS duration	40 μs
SIFS duration	10 μs	Transmission rate	6 Mbps
CW_0	4 slots	$CW_{\max}$	1024 slots
λ_A, λ_C	$\{50, 100, 200, 300\}$ frames/sec	Simulation time	10 sec

Table 1: Given Parameters

The following calculations show our conversion of the given parameters into slots:

-Each slot duration = 20 microseconds. Therefore, 1 second = $1 \div (2 * 10^{-5})$ = 50,000 slots

-Simulation time = (10 sec x 50,000 lots/sec) = 500,000 slots during the 10 seconds simulation

SIFS	**ACK, RTS, CTS [30 bytes =2 slots]**	**DIFS**	**CW *o***	**CW *max***
10 microsec	**40 microsec**	**40 microsec**	**80 microsecs**	**20480 microsec**
0.5 slot	**2 slots**	**2 slots**	**4 slots**	**1024 slots**

Table 2: Representation of Data in Slots

Code Model:

For data transmission, the behavior must go through each of these stages, if it does not, a collision has occurred.

CSMA/CA

```
For 1sec:50,000 slots
If (DIFs)
 node(t) = DIFS;
  Else if (CW counting down)
   node(t) = CW
    Else if (sending data)
      node(t) = DATA
       Else if (waiting SIFS)
          node(t) = SIFS
           Else if (ACK)
             node(t) = ACK;
Else (collision)
End
```

VCS

```
for 1sec:50,000slots
 If (sender node freeze)
   Start exponential backoff mechanism
    Else if (send RTS)
      node(t) = RTS
        Sensing collision;
       Else if (waiting for SIFS before CTS)
         node(t) = SIFS
          Else if (receiver node freeze)
             Start exponential backoff
mechanism
            Else if (receive CTS)
              node(t) = CTS
              Else if (wait for SIFS before
DATA)
                node(t) = SIFS
                    Else if (sending DATA)
                      node(t) = DATA
                       Else if (waiting SIFS)
                         node(t) = SIFS
                           Else if (ACK)
                             Node(t) =ACK
End
```

Using the pseudo code from above, our group applied the logic to MATLAB to simulate and generate the results of each node's throughput, delay, and number of collisions by plugging in the respective λ value.

Charts & Observations:

1. Throughput

Throughput is the rate at which data is transmitted. Our throughput value is calculated as follows:

Tt = (Successful sent frames * data frame size * 8) / (Last frame received - 1st frame transmitted)

Below are the charts that our group generated for the throughput:

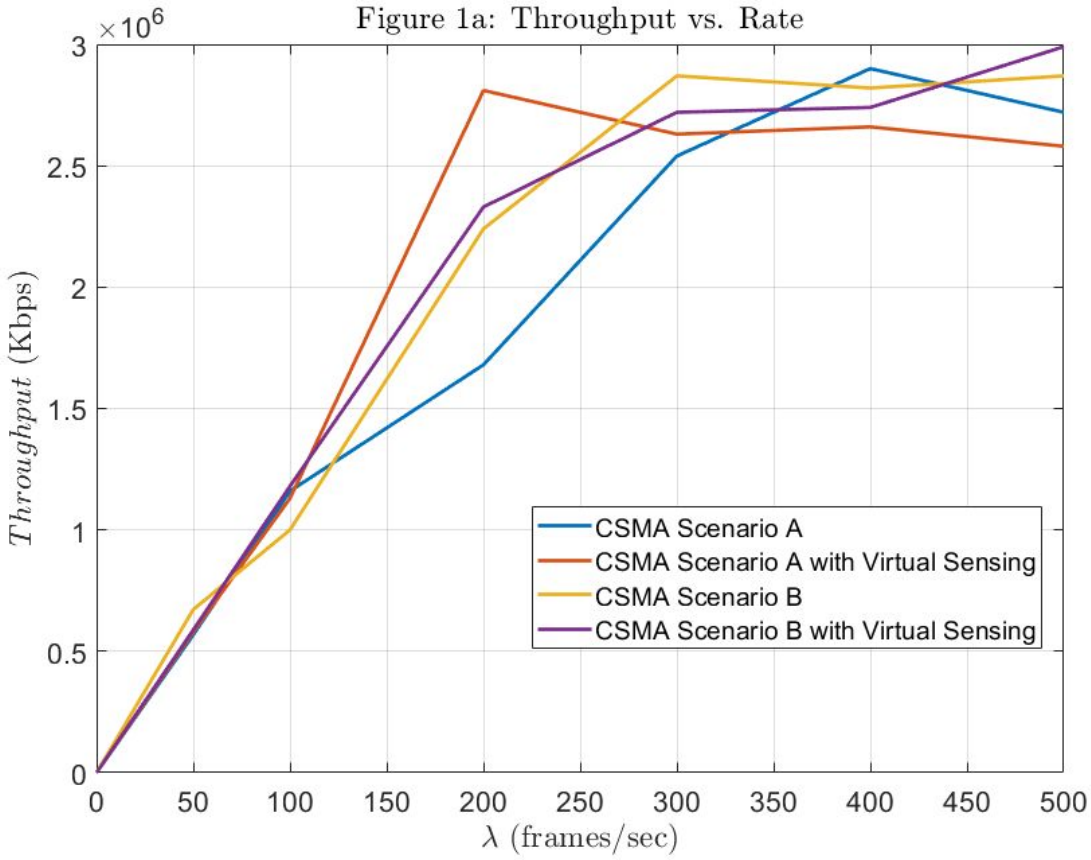

Figure 1a: Node A - Throughput T (Kpbs) vs Rate for Scenario A & B - CSMA Implementation 1 & 2

Observations:

In Figure 1a, we can see that Node A's throughput trend stays consistent for both Scenario A (Concurrent Communication) & B (Hidden Terminal) where the throughputs are all relatively high. Although, the max increment of the throughput did slow down around 400 frames/sec possibly due to an increase in collisions. In Scenario A, four nodes can sense one another in the network environment. Around 200 frames/sec, VCS seems to be doing better with the higher throughput than the CSMA/CA alone. However, at double the rate at 400 frames/sec, it seems as though all lines come to a similar throughput.

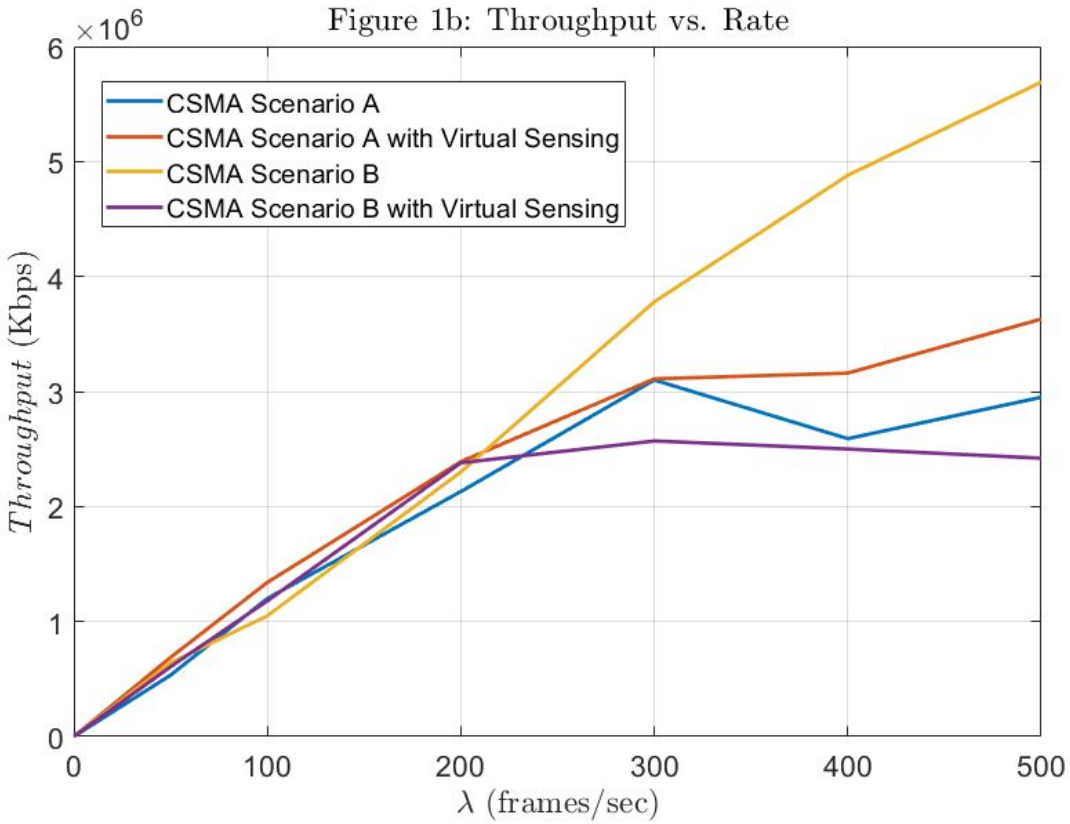

Figure 1b: Node C - Throughput T (Kbps) vs. rate for scenario A & B - CSMA 1 & 2

Observations:

For Node C, we can see an obvious increase in throughput for CSMA/CA in scenario B due to the node constantly trying to retransmit packets. There is no restriction from Node C to Node B (yellow line). However, after VCS is enabled, we can see a big difference in the decrease of throughput in the purple line. This is due to the fact that Node C can sense the hidden node (A) from Node B's CTS signal, and it will defer its transmission to avoid collision.

Without VCS enabled, Node C has an overall higher throughput in the CSMA/CA mode. With that, we should expect more collisions to occur from A causing no collision at C later due to sensing.

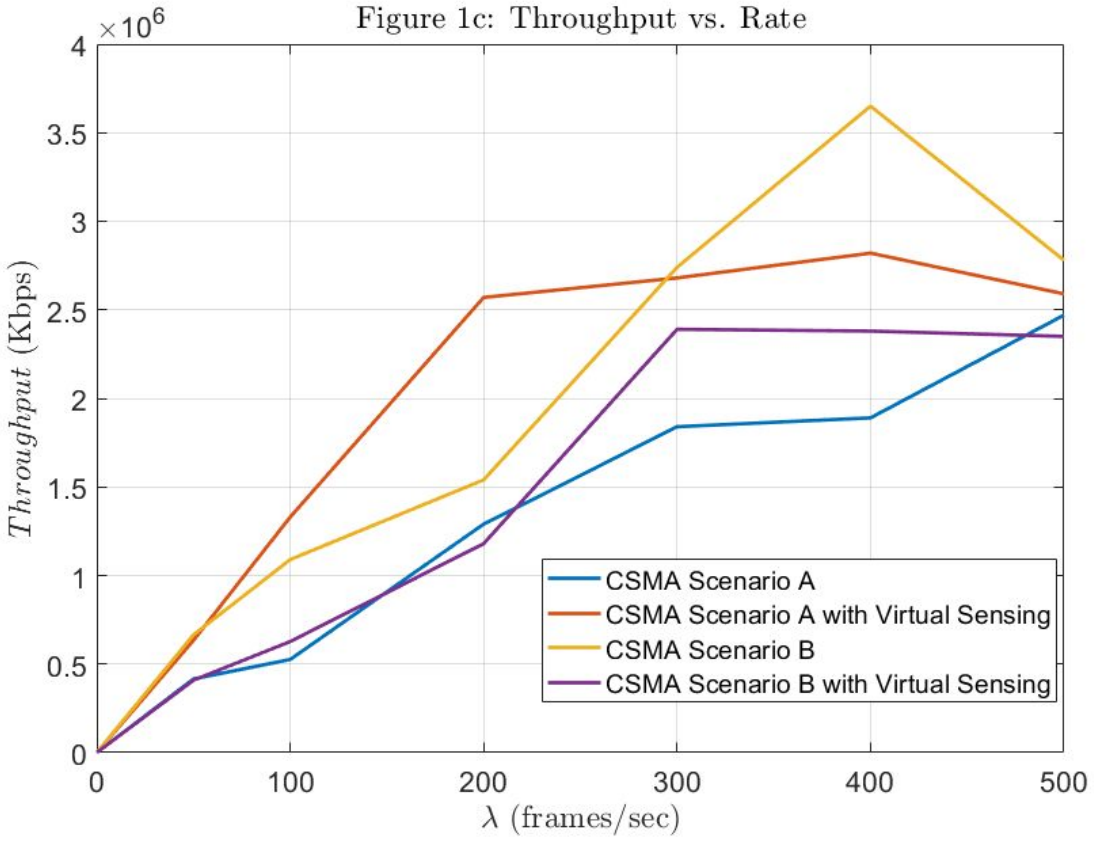

Figure 1c: Node A - Throughput T vs Rate for A & B - CSMA implementation 1 & 2 when $\lambda A = 2\lambda C$

Observations:

When $\lambda A = 2\lambda C$, the curve of Node A in both scenarios appears to be similar despite whether or not VCS was enabled.. A potential reason for this is due to CTS/RTS trying to sense the channel, but the channel appears to be always idle or busy. Using Virtual Sensing with CTS/RTS does not help with an increasing throughput at a larger rate in this case.

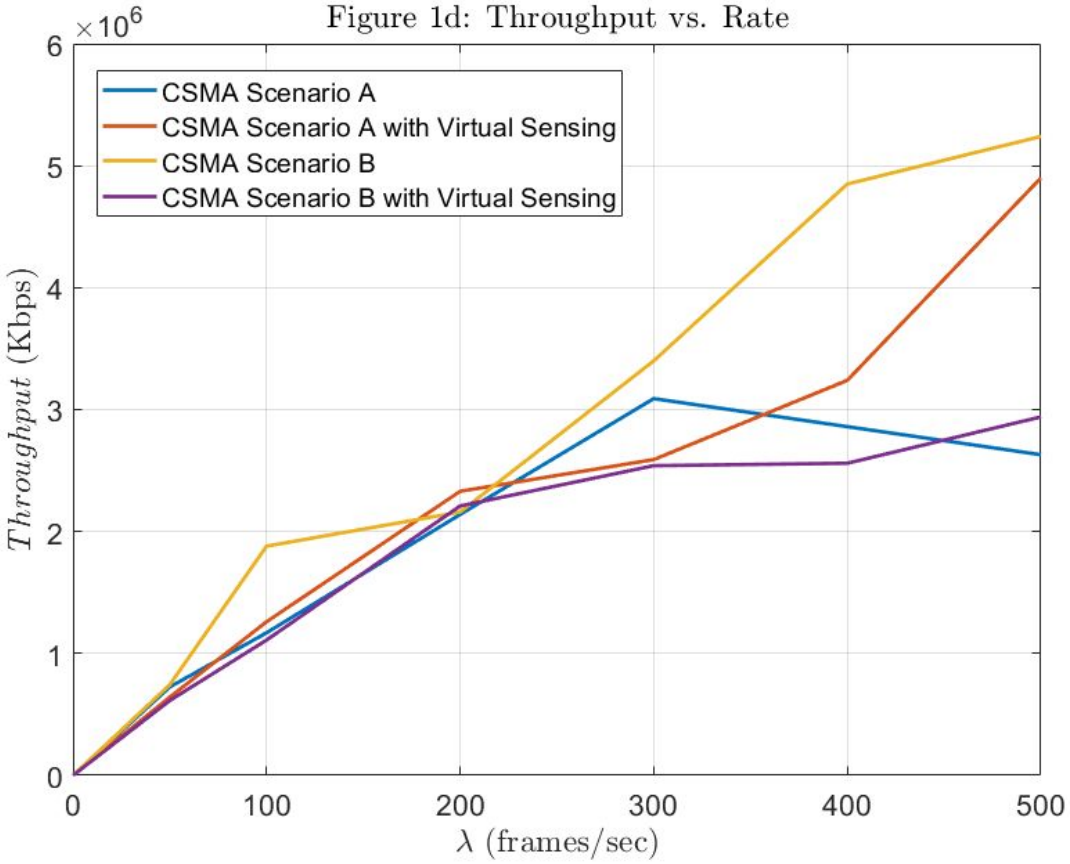

Figure 1d: Node C - Throughput T vs. rate for A & B - CSMA 1 & 2 when $\lambda A = 2\lambda C$

Observations:
We noticed a similar trend where Node C's throughput in Scenario B is the highest, but completely lowered once Virtual Sensing was enabled.

As for Scenario A, Node C's throughput was steady until it reached 300 frames/sec and started to decrease. When Virtual Sensing was enabled, the throughput actually kept increasing.

2. Number of Collisions

1a. CSMA/CA - Collision@ Node B & D when A & C transmit concurrently (not using VCS)
1b. VCS - Collisions @ Node B & D when Node A & C broadcast RTS to B & D at the same time. Node A & C = double contention window and select another backoff time randomly.

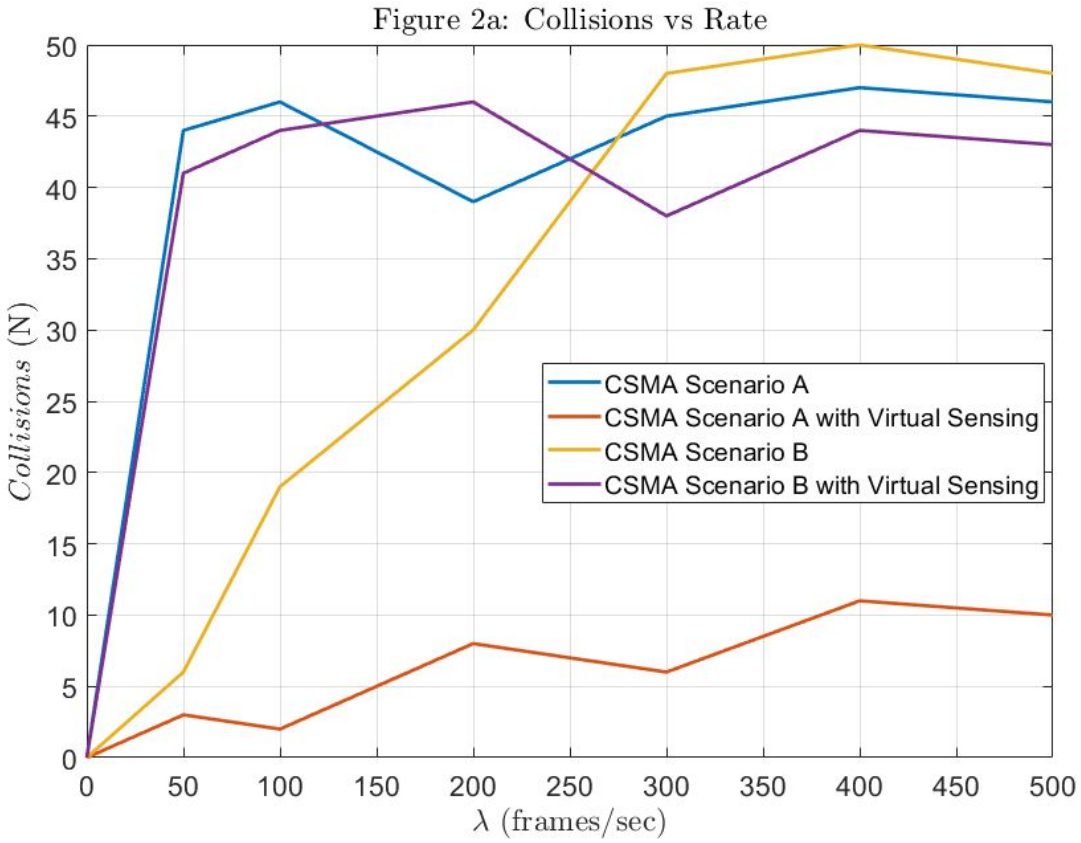

Figure 2a: Node A - # Collisions vs Rate for A, B - CSMA 1 & 2

Observations:

For Scenario A, Node A's collisions were the lowest when virtual sensing was enabled. Collisions were below 15 at all times for the rates ranging from 50-500 frames/sec.

For Scenario B, collisions were the highest in the CSMA/CA case at a steady rate. There was a directly proportional relationship between the rate and the number of collisions. Essentially, the higher the rate, the more collisions occurred.

CSMA/CA and CSMA/CA with VCS Scenarios A and CSMA/CA Scenario B with VCS are high because the node will keep sending packets.

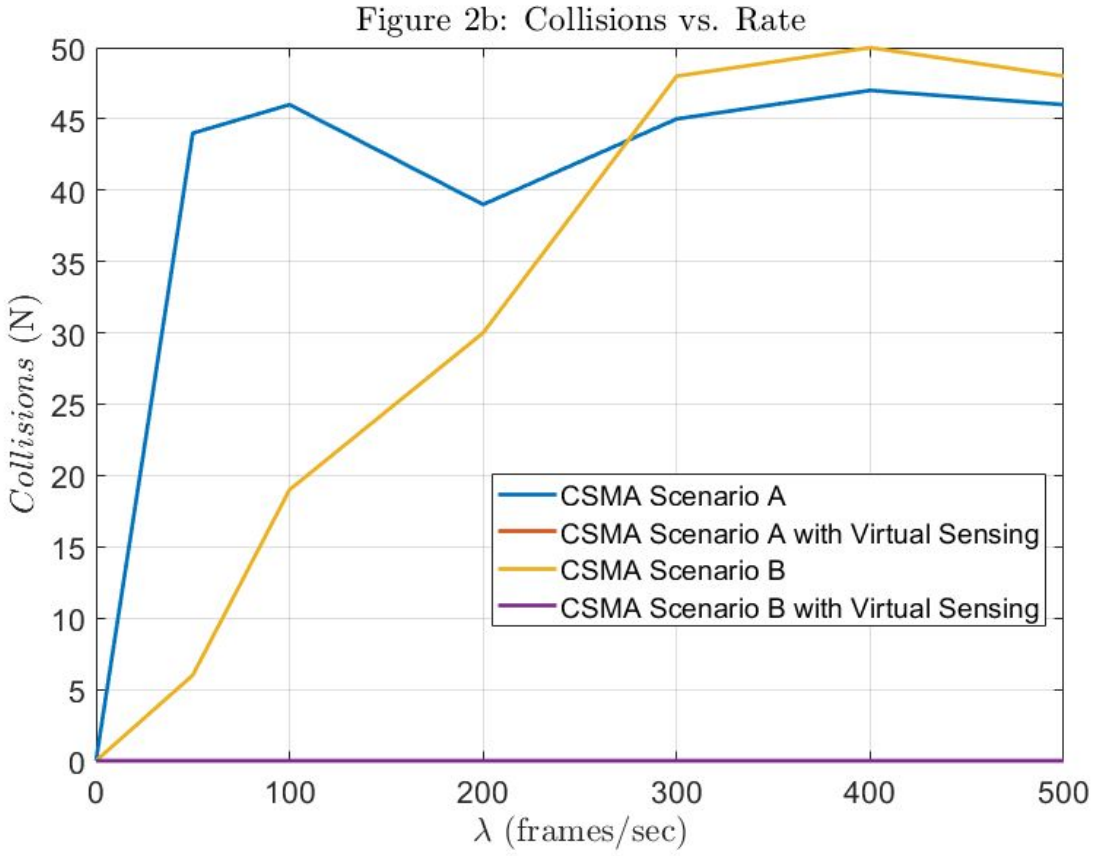

Figure 2b: Node C - # Collisions vs Rate for A, B - CSMA 1 & 2

Observations:

In this chart, we noticed that CSMA/CA Scenario A with VCS and CSMA /CA Scenario B with VCS have no collision. The reason for this is due to the fact that the node always senses that the channel is busy or idle. Node C will always wait or backoff before sending a packet.

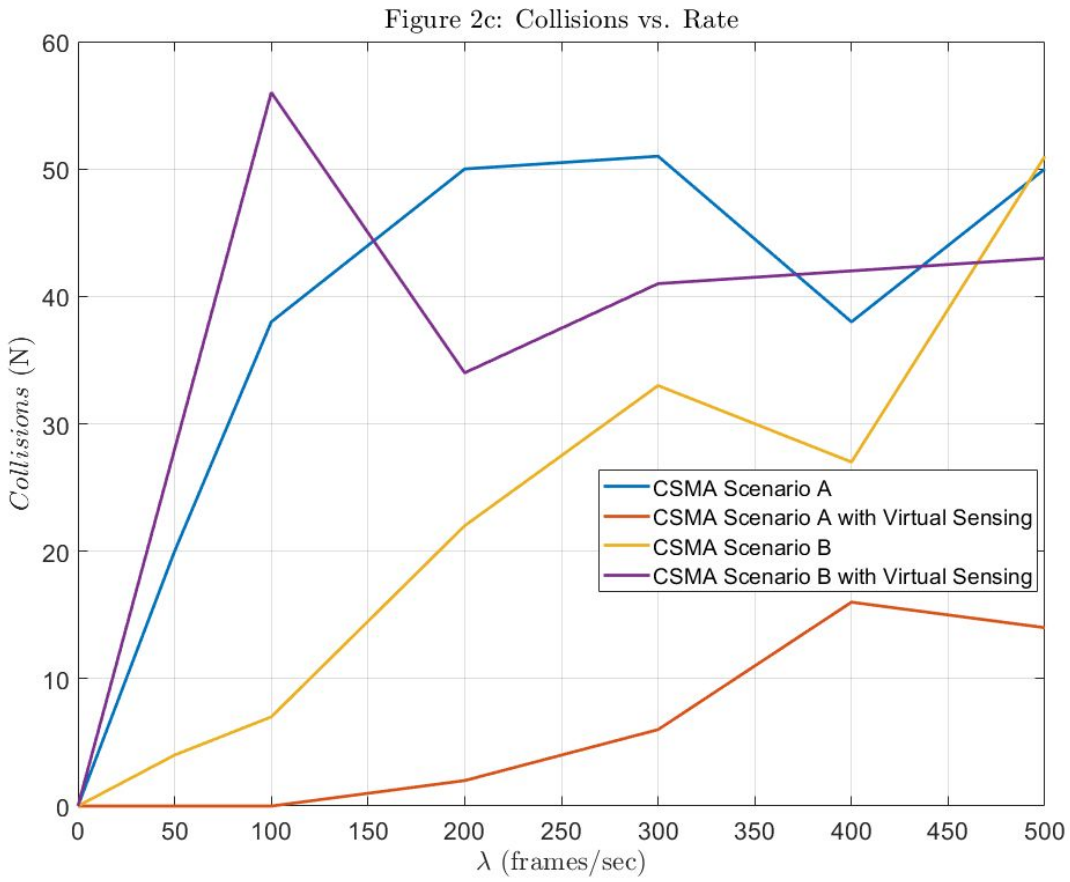

Figure 2c: Node A - # Collisions vs Rate for A, B - CSMA 1 & 2 when $\lambda A = 2\lambda C$

Observations:

The CSMA/CA Scenario A with Virtual Sensing had the least amount of collisions due to the fact that Node A or Node B will always sense a busy or idle channel signal. Therefore, either node will wait and backoff before transmitting. One can compare the purple line and yellow line for Scenario B - where the collision rate increases as the rate increases, and with VCS enabled, initial collisions are high but eventually drop to a lower value. This is also a similar trend to Figure 2a.

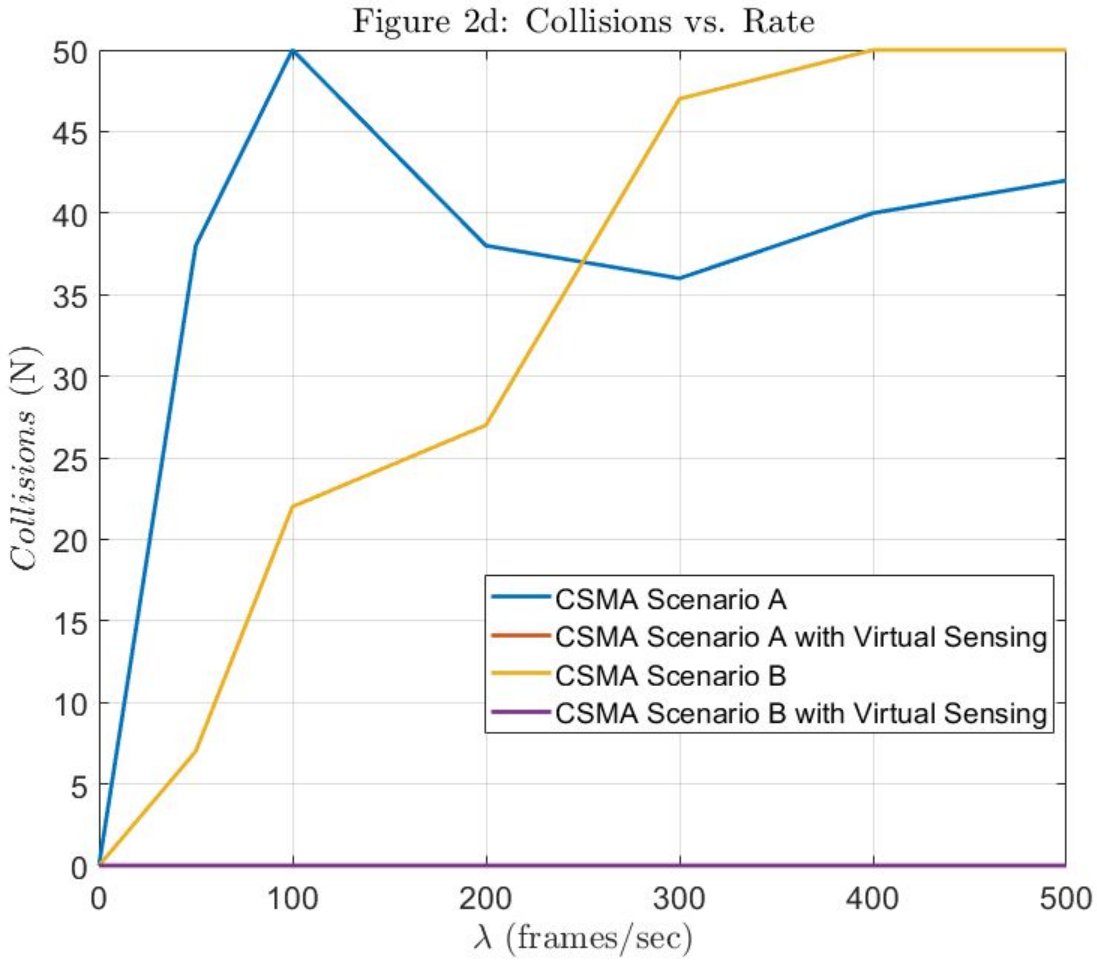

Figure 2d: Node C - # Collisions vs Rate for A, B - CSMA 1 & 2 when $\lambda A = 2\lambda C$

Observations:

CSMA/CA Scenario A & B with VCS have no collision. This is due to the fact that it always senses the channel is busy or idle, so Node A & C will always wait or backoff before sending a packet.

3. Fairness Index

The fairness index (FI) represents the ratio of the throughput on node A to the throughput of node C. The closer the FI value is to 1, then the more fair the throughput is for both channels.

The fairness index is calculated using the following equation:

$$f(x_1, x_2, \cdots, x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2}$$

Where n represents the number of nodes available and x_i is the ratio of actual throughput to optimal throughput. The values of x_i needed to be normalized by the transmission rate.

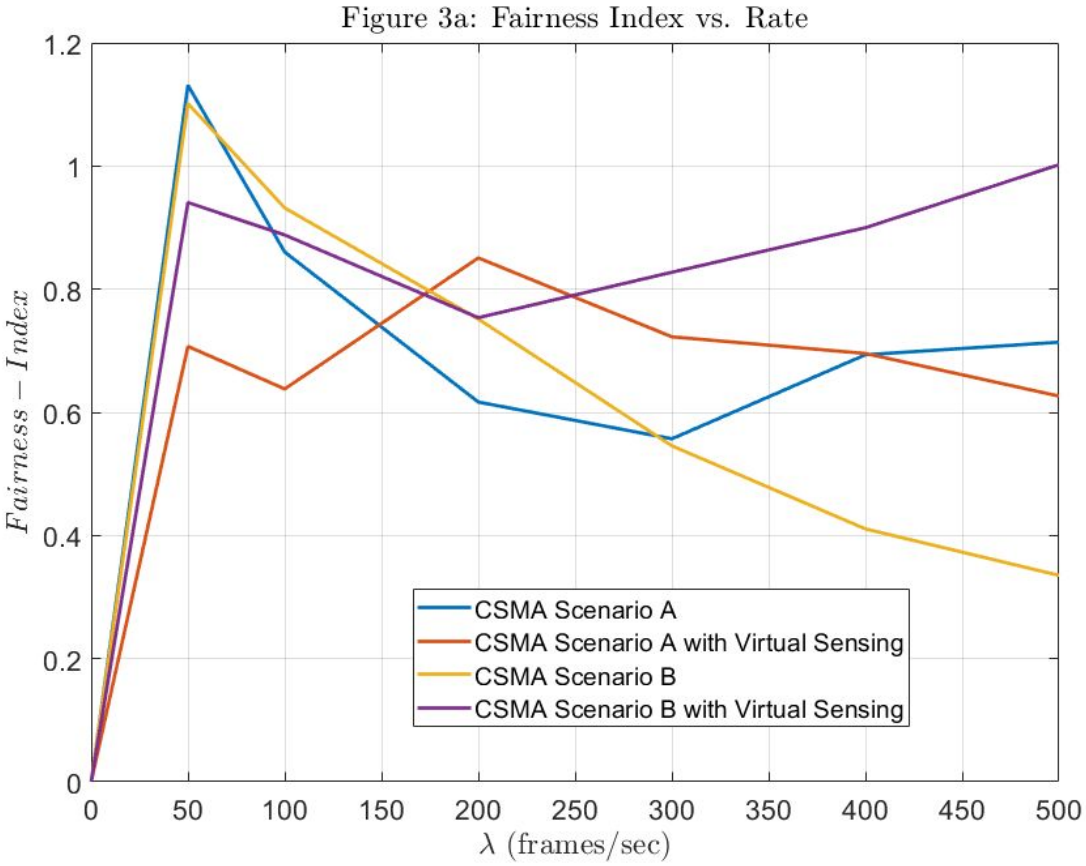

Figure 3a: FI vs. Rate for scenario A & B

Observations:

In this chart, we notice that there is a downtrend of all the curves. CSMA/CA appears to decrease quicker as λ increases. Out of all the charts, Scenario B with Virtual Sensing produced the most fair ratio out of all the line scenarios with a value much closer to 1.

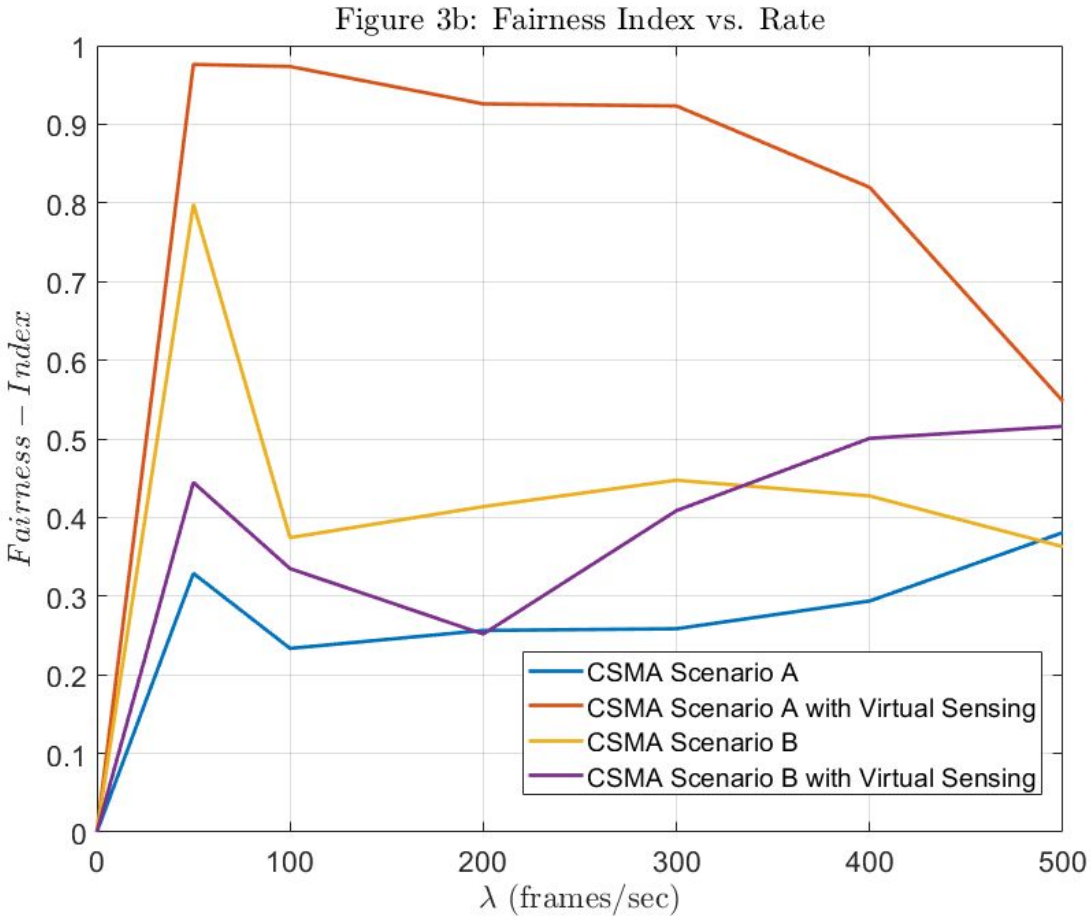

Figure 3b: FI vs. Rate for scenario A & B when $\lambda A = 2\lambda C$

Observations:

When $\lambda A = 2\lambda C$, each of the scenarios started much closer to 1 with the lower λ value. As the λ increased, the further the value shifted away from 1. We can see this correlation in Scenario A with VCS enabled - the FI value started off much closer to 1, and as frames/sec increased, the fairness decreased. As λ increases, Node C has more opportunities to transmit frames, but it also has more potential to create collision at Node B, which affects the FI value. However, with the VCS implementation, the correlation trend starts to increase again as seen in the purple line.

Conclusion:

Overall, this project helped our group to gain a greater understanding and knowledge of the practical approach of wireless communications protocols and packets transmission by being able to apply and analyze how rates, throughput, collisions, and the statistical method of fairness index come into play with different network topologies. Our group created two theoretical coding structures to model the behavior of the two different network topologies. The code was created in MATLAB and later ran with discrete simulation parameters. The results were graphed

and then analyzed for correctness and overall behavior based on theory learned in the course and independent research. Through our data, our group concluded enabling Virtual Carrier Sensing provided much more applicable results in relation to network efficiency.