Cody Stein

Finn Willard

October 11th 2020

ECE 578 - Project 1

1. Introduction

This report studies and presents the results of two 802.11 DCF network topologies with two network access protocols. The first topology, seen in Figure 1a, and will be further be referred to as scenario A, is a scenario in which parallel transmissions from sender A and C share the same collision domain. The second topology, seen in Figure 1b, and will further be referred to as scenario B, is similar to scenario A with exception that the senders, A and C, are hidden terminals. Both scenarios will be evaluated against one another by examining the performance of Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) and Carrier-Sense Multiple Access with Virtual Carrier Sensing (CSMA VCS) network protocols.

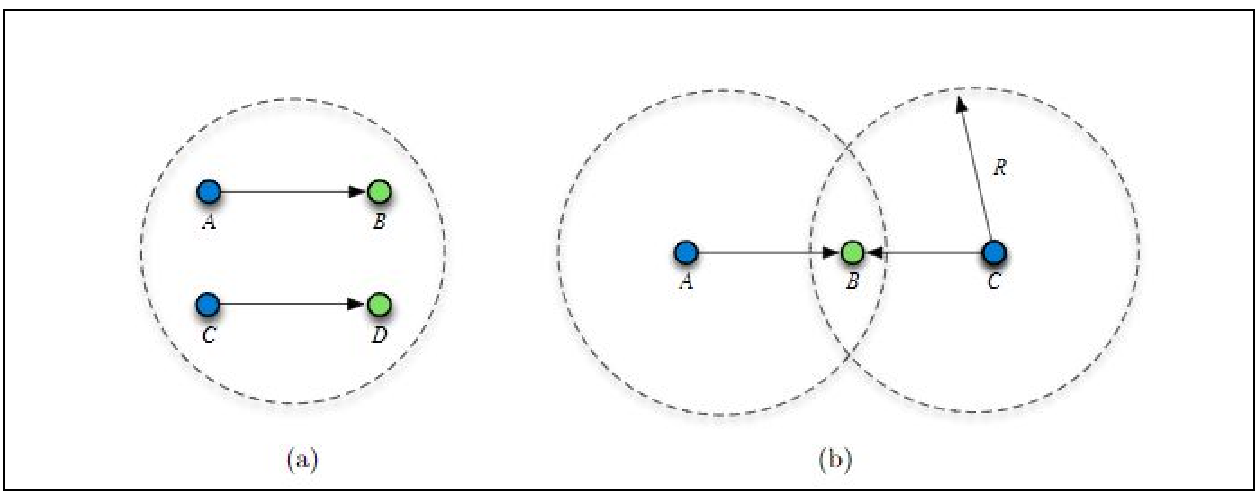


Figure 1

Throughout this project Cody and Finn worked together to develop the simulation with Finn making initial commits and Cody addressing bugs and validating the order of events in the event based simulation. Once the simulation reached a reasonable amount of integrity, Cody added methods to ascertain performance data as well as plotting methods to present the simulation outcomes. Development of the report consisted of Cody creating the outline and rough draft, peer reviews from Finn, and finally analysis of presented data was performed by Cody and Finn working together.

2. Simulation Development

Using python, a hybrid of a time based and event based simulation was developed. In this sense, the simulation steps through each slot and various slot timers indicate when events occur. The parameters used in this simulation are listed in Table 1.

Simulation Parameter	**Value**
Data frame size (bytes)	1500 bytes
Data frame size (slots)	50 slots
Slot duration	10 μs
SIFS duration	1 slot
CW_0	4 slots
$\lambda_A = \lambda_C$	[200,300,500,1000,2000] frames/seconds
ACK slot size	2 slots
RTS slot size	2 slots
CTS slot size	2 slots
DIFS duration	2 slots
Transmission rate	24 Mbps
CW_{max} size	1024 slots
Simulation time	10 seconds

Table 1

Queueing was done to schedule transmissions. Transmissions were scheduled exponentially using equation 1.

$$Equation\ 1:\ X\ =\ ln(1-U)\times(1\div\lambda)$$

In this case, X is representing the slot in which a transmission is added to a queue, λ represents the frame rate selected, and U represents a series of uniformly distributed numbers between 0 and 1.

The collision window (CW) was calculated by taking a random integer between 0 and an upper value determined using equation 2.

$$Equation\ 2:\ 2^{(k)}\ \times CW_0$$

All random number draws used throughout this simulation were seeded to ensure repeatable results.

The repository for the simulation can be found at:

https://github.com/handfulofsharks/ece578-proj1

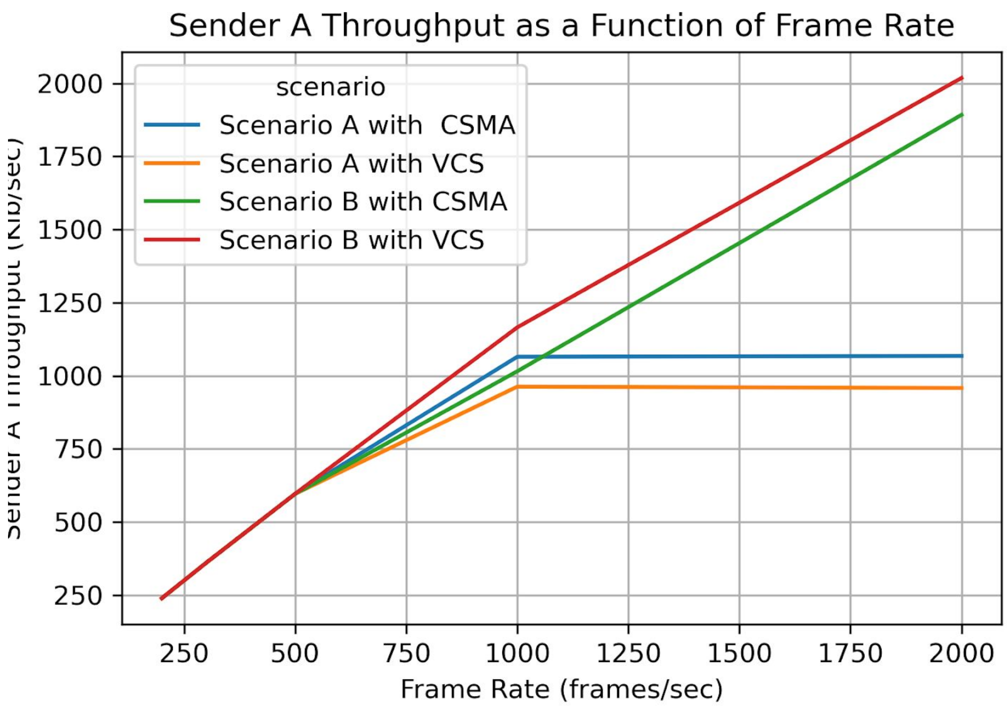
3. Analysis

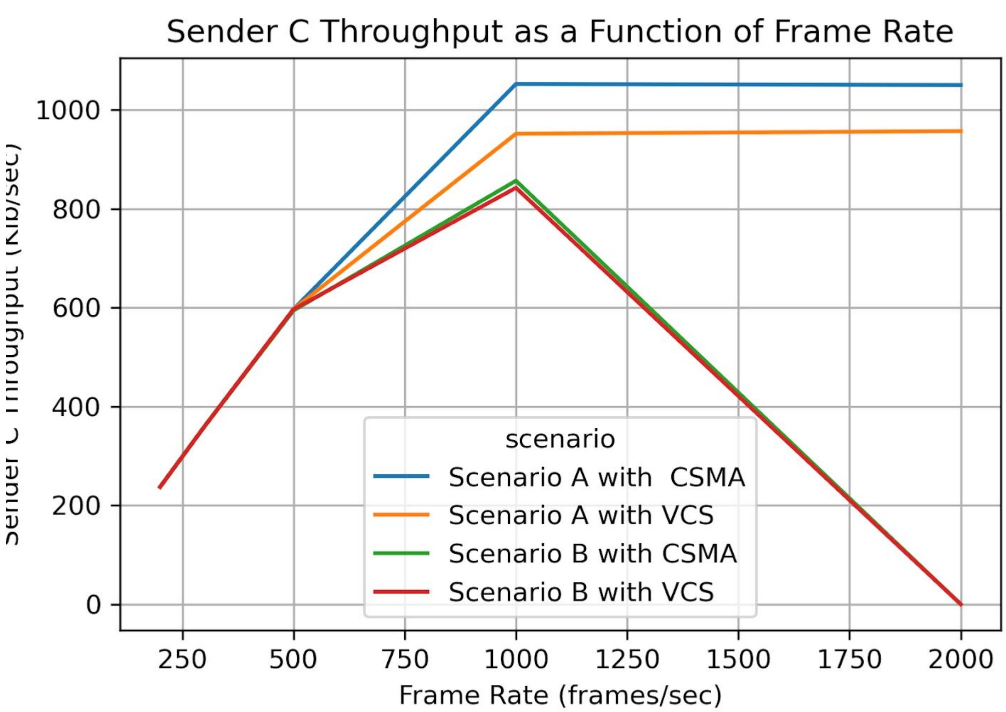
a. Throughput

Using *Equation 3*, throughput was calculated at the end of each frame for each scenario.

$$Equation\ 3:\ T\ =\ \frac{8\times Number\ of\ successes\times Data\ Frame\ size}{(Simulation\ time}$$

Transmissions were considered successful only when an acknowledgement was received by the sender. Partial transmissions under this criteria are not considered for this metric.

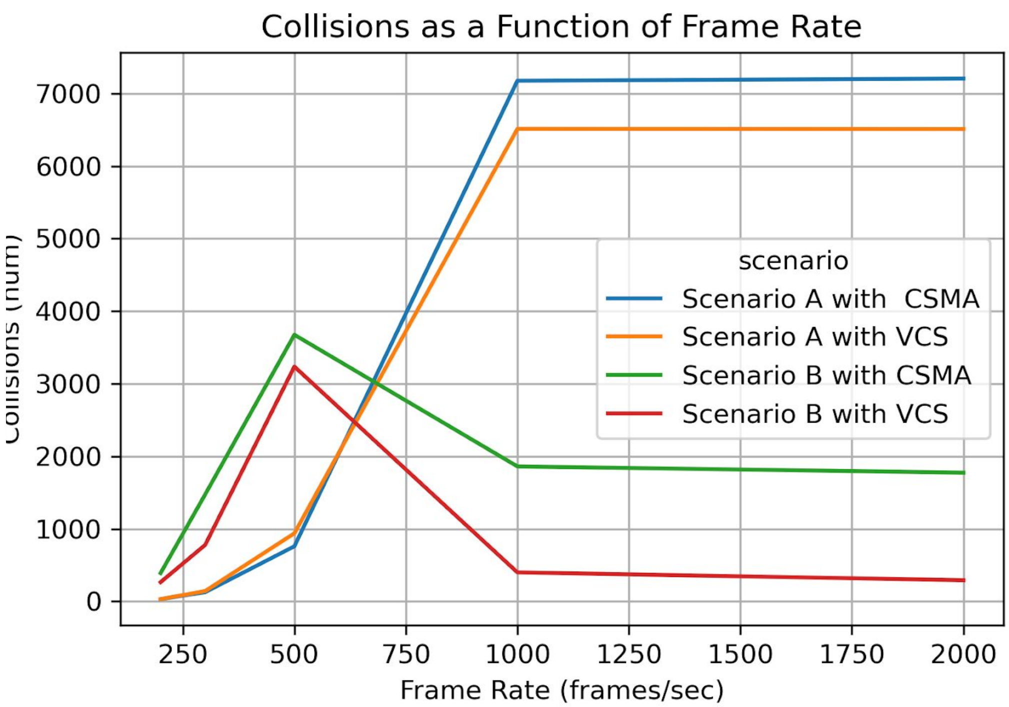




As can be seen in the figures above, when enabling VCS on Scenario A, the effect is simply a lowering of the throughput for both nodes. Since both transmitters are in the same collision domain, they are always able to detect when the channel is idle, and therefore don't need virtual carrier sensing to prevent collisions. Since each transmitter is able to tell when the other is occupying the channel, the need for RTS and CTS transmissions is eliminated. Collisions only happen when the two stations attempt to transmit on the exact same slot, and the net effect is a lowering of throughput by including the extra slots for the RTS and CTS transmissions. For scenario B our throughputs remain roughly the same until the frame rate rises above 500. With VCS enabled, the throughput for transmitter A is higher than C for higher frame rates, while the throughput for transmitter C remains roughly the same with or without VCS enabled. Additionally, as the frame rate increases, transmitter A begins to dominate the total throughput, with C dropping to near zero at the 2000 frames/second level. However, we believe that this may be due to a bug in our simulation that we were never able to nail down. When running our simulation from 200 frames/second to 2000 frames/second, we find that one transmitter consistently dominates the throughput, however the transmitter that dominates changes depending on the random generator aspects of the simulation.

b. Collisions

Collisions are determined by evaluating whether or not the node states are both in the ‘transmitting’ state at the same time.

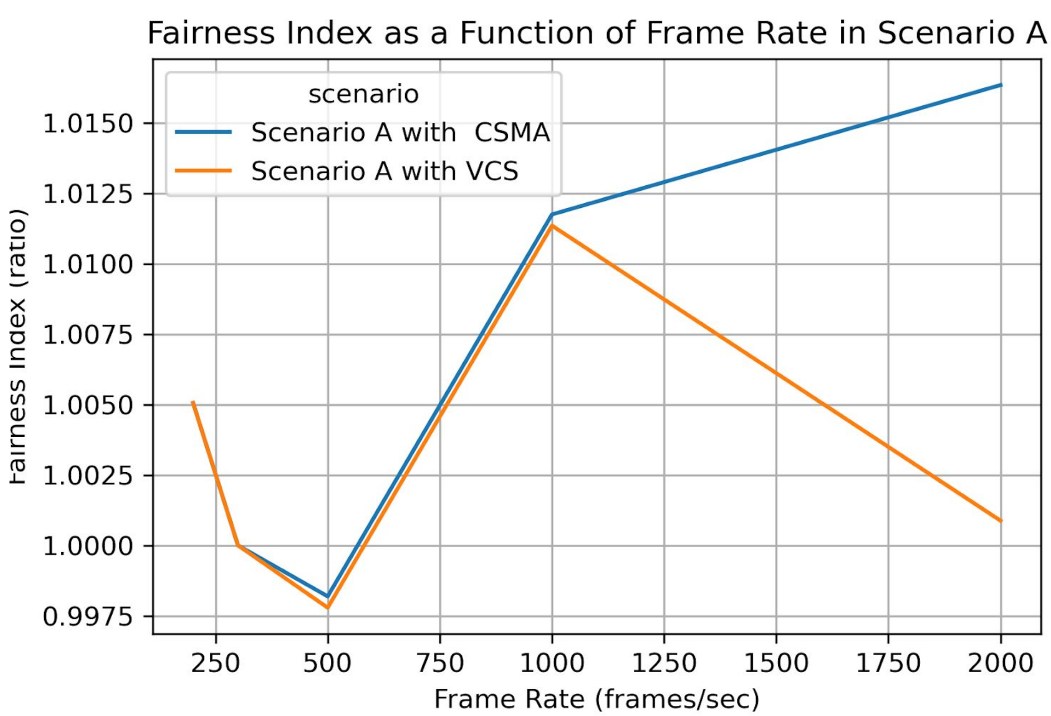


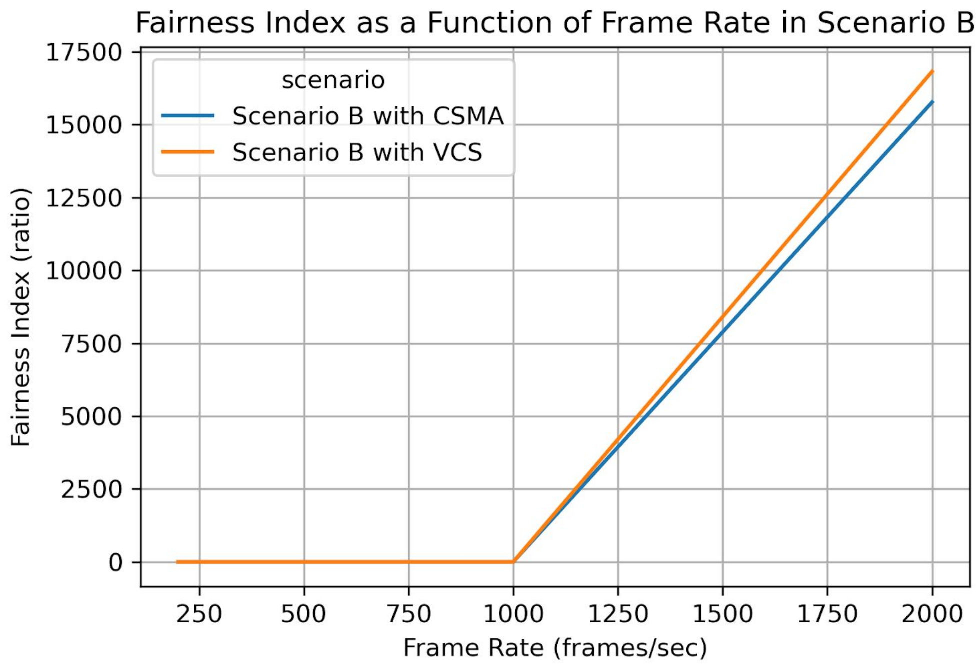
In the case of scenario A, when both transmitters are within the same collision domain, each transmitter is able to sense when the channel is busy without the potential for hidden collisions. Since both transmitters can see when the other is transmitting, the only time collisions occur is when the two transmitters attempt to transmit in the same slot. This should theoretically result in a lower collision rate in comparison to scenario B. However, our simulation resulted in a lower collision rate for scenario B when the frame rate rose above 750 frames/second. We believe that this is the result of a bug in our simulation, but we were never able to pinpoint the exact cause of this discrepancy. It stands to reason, however, that the hidden terminal (scenario B) would result in a higher number of collisions, since each transmitter cannot as accurately assess when the channel is idle. For scenario A, when virtual carrier sensing is enabled the number of collisions remains roughly the same for frame rates below 1000, and then is slightly lower for the remaining frame rates. In scenario B, the virtual carrier sensing allows the hidden terminals to know when the receiver is busy through the NAV. This results in the number of collisions being lowered regardless of the frame rate.

c. Fairness Index

Fairness index is a representation of how "fair" the network treats '*n'* amount of users. Using equation 4, the Fairness Index was calculated for each of our scenarios at each frame rate.

$$Equation\ 4:\ FI\ =\ \frac{(\#\ of\ successes\ for\ A \times frame\ slot\ duration) \div total\ amount\ of\ slots}{(\#\ of\ successes\ for\ C \times frame\ slot\ duration) \div total\ amount\ of\ slots}$$





As seen in scenario A, the fairness index fluctuates, but very minutely, hovering around 1. This would indicate there is a slight bias towards sender A but more than likely within a negligible amount.

However, when examining scenario B and taking into consideration our theory that oscillations of high and low throughput occur between senders A and C, polarizing fairness indices were produced. Given a certain frequency and seed, our simulation indicates that scenario A produces a much more consistent and reliable fairness index throughout the spectrum of frequencies. In contrast, scenario B produces a consistent fairness index up to around whereas scenario B is consistent up to around 1000 frames per second. After which, the fairness index indicates a heavy lien towards one receiver depending on the oscillation you are measuring.