Sheldon Ruiz

ECE 478

Project 1

Introduction

Problem Statement

The Purpose of this project was to simulate the CSMA/CA and CSMA/CA with VCS (virtual carrier sensing) protocols in an 802.11 DCF wireless setting. These simulations were swept across two different collision domains and through two different frame rate sets (each set containing four frame rates). For the traffic, a Poisson Distribution was generated at intervals in accordance with the frame rate and desired simulation time. The following simulation parameters were used to build this project:

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} frame/sec	Simulation time	10 sec

Table 1: Simulation Parameters

Simulating the problem description required two different simulation types and two different simulation domains over the frame rate sets. These simulations were CSMA/CA Single [collision domain], CSMA/CA Hidden, CSMA/CA + VCS Single, and CSMA/CA + VCS Hidden. The first of these simulation domains, Single, described the situation where four stations, two receivers and two transmitters, occupied the same collision domain:

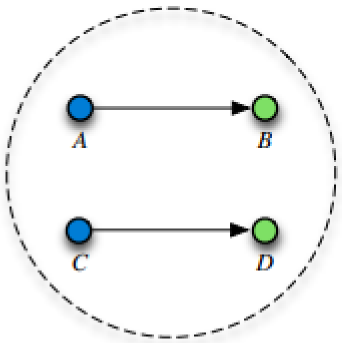


Figure 1: *Single Collision Domain (figure courtesy of Project Manual)*

The CMSA/CA and CSMA/CA+VCS protocols were simulated in this collision domain for a scenario A with frame rate set $\lambda_A, \lambda_c =$ {50, 100, 200, 300} and scenario B with the frame rate set doubled for sender A ($2\lambda_A$).

The Hidden collision domain description:

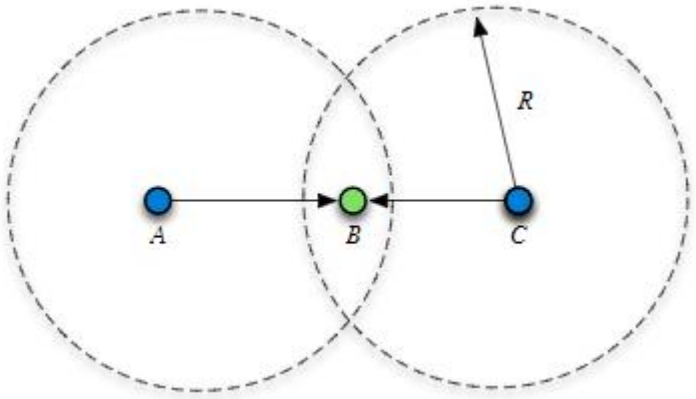


Figure 2: *Hidden Collision Domain (figure courtesy of Project Manual)*

In the Hidden Collision domain, B was the shared received between senders A and C. A and C cannot "hear" the signals propagating from the other. In this domain CMSA/CA and CSMA/CA+VCS were both also simulated over scenarios A and B as described above.

Responsibilities

This project was worked on in conjunction with finding a team mate to meet the deadline. Unfortunately, the project was completed before finding a team mate. Thus, all work was developed by Sheldon Ruiz.

Methods

Functionality Overview

Java was used to create an Object Orientated (OO) approach to the system. Stations, Channels, Simulations, and Packet Generators were designed as classes.

Stations contained information about the transmitting state of the station, the status of the station, its Poisson Distributed traffic, and the simulation parameters. Additionally, the station class encompasses status update logic processes relevant to each simulation type.

Channel(s) represented the link between a sending and receiving station. A channel had an owner (the occupying station), a busy status, and a record of the time spent connected to each station.

Packet Generators contained a simple static method callable by the stations to generate their respective traffic. The traffic was generated through the following methods. First, a series of uniformly distributed random numbers (set U) between 0 and 1 were generated as a seed to the exponentially distributed traffic series (X). This can be described as:

$$U = \{u_1, u_2, \dots, u_n\} \mid u_i \epsilon (0, 1)$$

$$X = -\frac{1}{\lambda(transfer\ rate)} * \ln(1 - U)$$

Next, the elements of the X set were shifted from the time domain to the slot domain. Finally, the elements of X were reassigned as the sum of the previous elements:

$$X_i = \frac{X_i}{SlotDuration}$$

$$X_i = \sum_{0}^{i} X_i$$

Simulations contained information relevant to executing the simulation of each type and domain. The class contains processes in setting up and simulating each protocol and each domain over the frame rate scenarios A and B. The simulation process for each type handles the difference between single and hidden domains.

Results

Throughput (T)

a. Node A, collision domains A and B, frame rate scenario A:

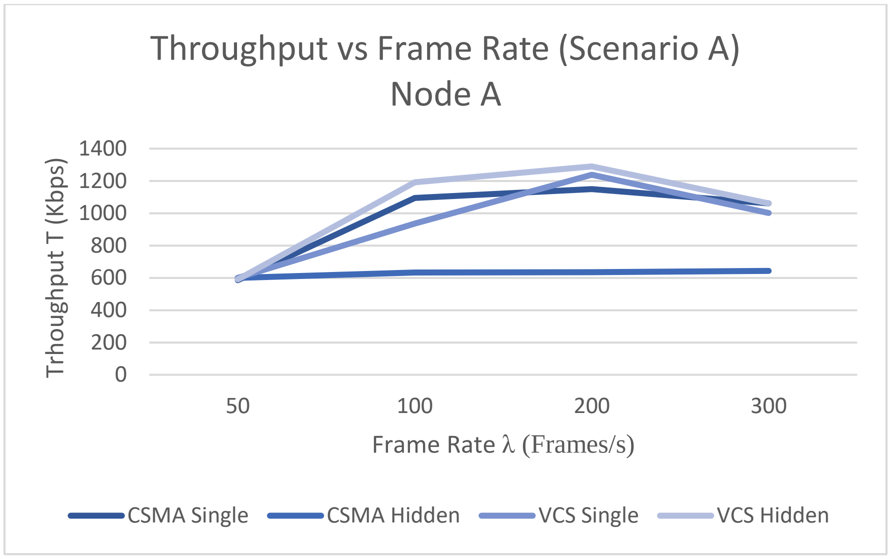


Figure 3: *T vs Frame Rate for all types, frame rate scenario A, node A*

Here, all protocols and domains start roughly equal, as the frame rate is low enough relative to the total simulation time that collisions are not frequent. As the framerate increase, throughput increase until a saturation point (100 – 200 fps), and then further framerate increases collisions. This is decrease in T is observable past 200 fps. CSMA Hidden contains the lowest throughput by far as collisions between packets are frequent with no VCS and in a hidden domains setting. VCS Single and VCS Hidden both max out at a higher rate than CSMA single, as the safeguards decrease packet collisions at high framerates.

b. Node C, collision domains A and B, frame rate scenario A:

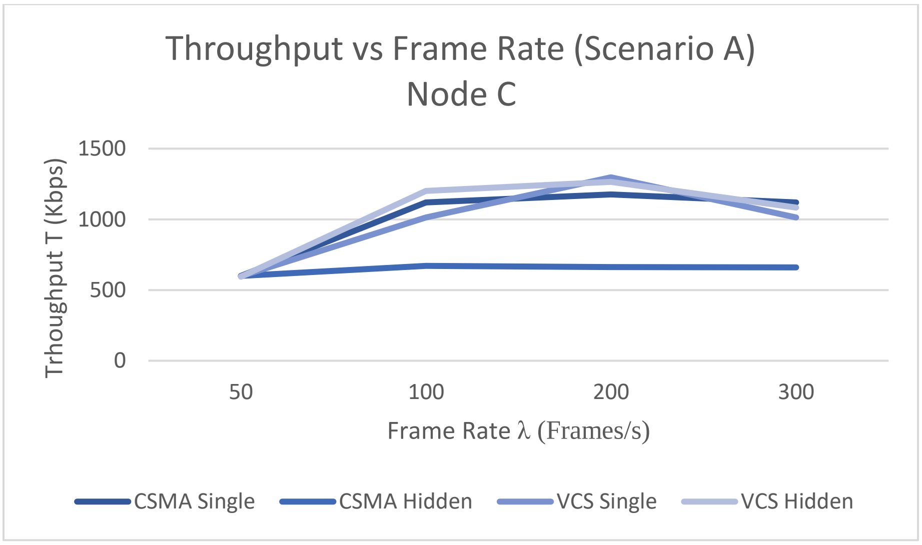


Figure 4: T vs Frame Rate for all types, frame rate scenario A, node C

The analysis here mirrors that of a) as there is no differences in simulation between node A and C. Any differences in throughput is due to the randomness of the collisions.

c. Node A, collision domains A and B, frame rate scenario B:

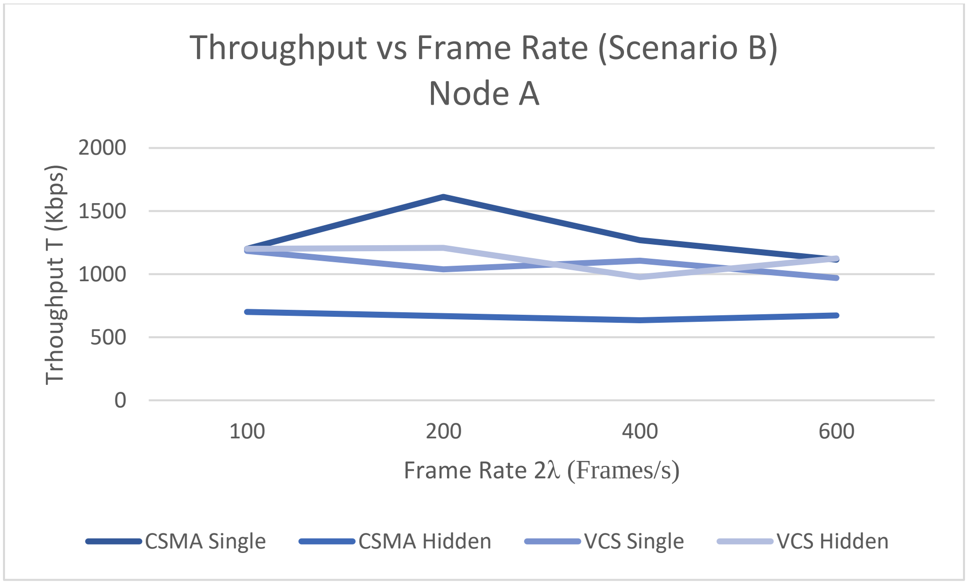


Figure 5: *T vs Frame Rate for all types, frame rate scenario B, node A*

In this frame rate scenario, Node A's traffic frame rate is doubled, yet Node C's remains the same. This allows A to achieve saturating throughput quickly with less collisions than if C's traffic were also doubled, resulting also in higher throughput. However, the throughput still caps at about 200 fps and descends afterwards.

d. Node C, collision domains A and B, frame rate scenario B:

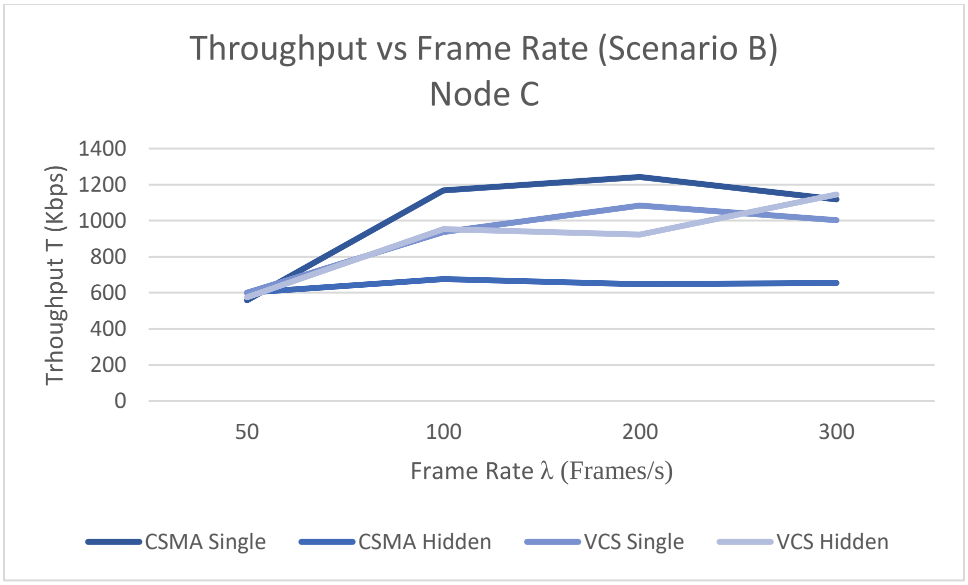
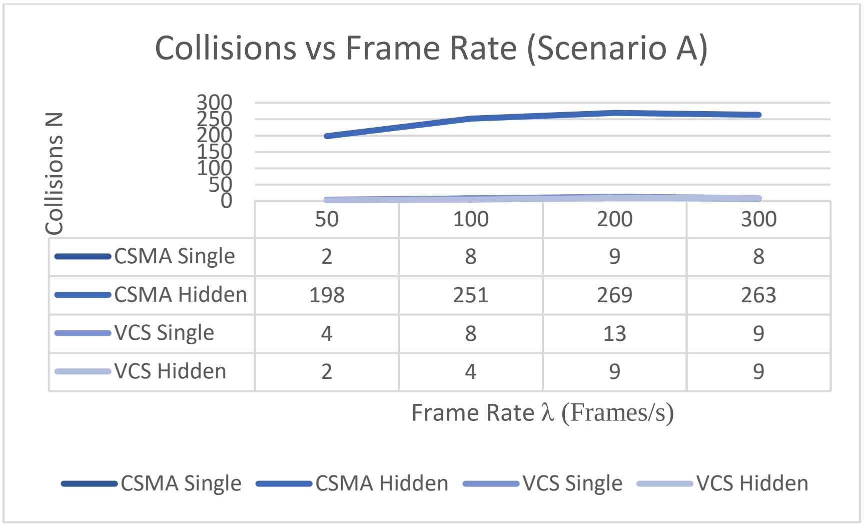


Figure 6: *T vs Frame Rate for all types, frame rate scenario B, node C*

These simulations are similar to throughput case b). The difference is now lesser maximum throughput for Node C as Node A is transmitting packets at double the frame rate

Collisions N

a. Node A (and C), collision domains A and B, frame rate scenario A:

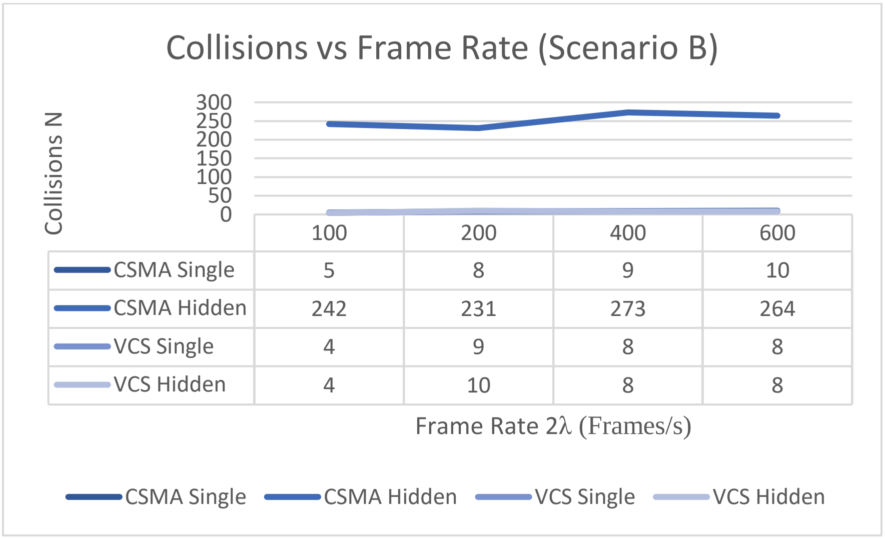


	50	100	200	300
CSMA Single	2	8	9	8
CSMA Hidden	198	251	269	263
VCS Single	4	8	13	9
VCS Hidden	2	4	9	9

Figure 7: *N vs Frame Rate for all types, frame rate scenario A, node A (and C)*

The collisions between A and C's traffic is represented here as collisions for each simulation type, per frame rate. Notice how CSMA Hidden reaches a substantially higher collision rate than the other protocols/domains. This is because CMSA without VCS in a hidden collision domain setting provides for no safeguards against collision – the stations cannot listen to the line correctly.

b. Same as a), collisions are recorded between A and C's packets.
c. Node A (and C), collision domains A and B, frame rate scenario B:



	100	200	400	600
CSMA Single	5	8	9	10
CSMA Hidden	242	231	273	264
VCS Single	4	9	8	8
VCS Hidden	4	10	8	8

Figure 8: *N vs Frame Rate for all types, frame rate scenario B, node A (and C)*

With A transmitting twice as many packets per second as C, the chance for collision is higher. This is observed in the data as on average each protocol saw in increase in collisions. This is especially prevalent in the CSMA Hidden simulation.

d. Same as c).

Fairness Index FI

a. FI, both collision domains, frame rate scenario A

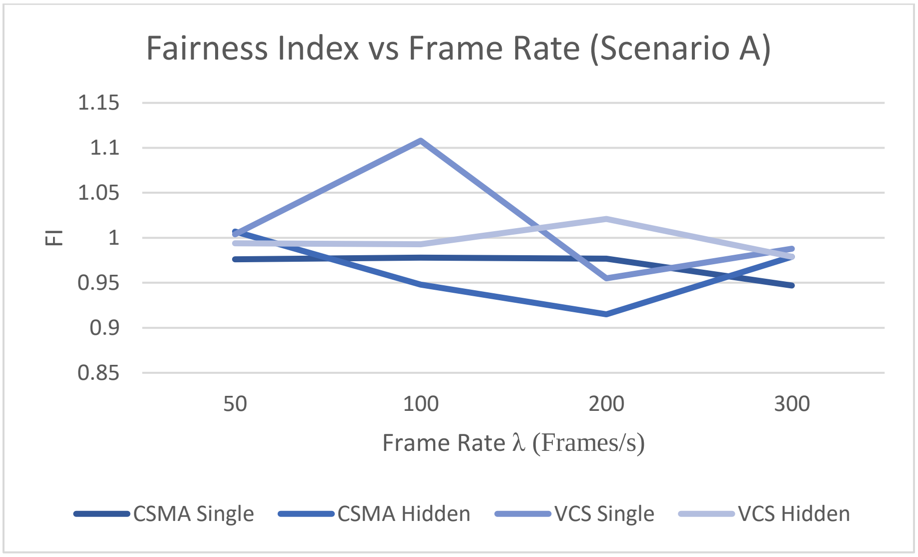


Figure 9: *FI vs Frame Rate for all types, frame rate scenario A*

The Fairness Index (FI) is entirely dependent on the ratio time the receiver spends with A's packets over the time spent on C's packets. In the equal frame rate scenario this is random as the both stations A and C transmit at the same rate, and receiver access is randomly given to the station that is awarded the lower backoff time.

b. FI, both collision domains, frame rate scenario B

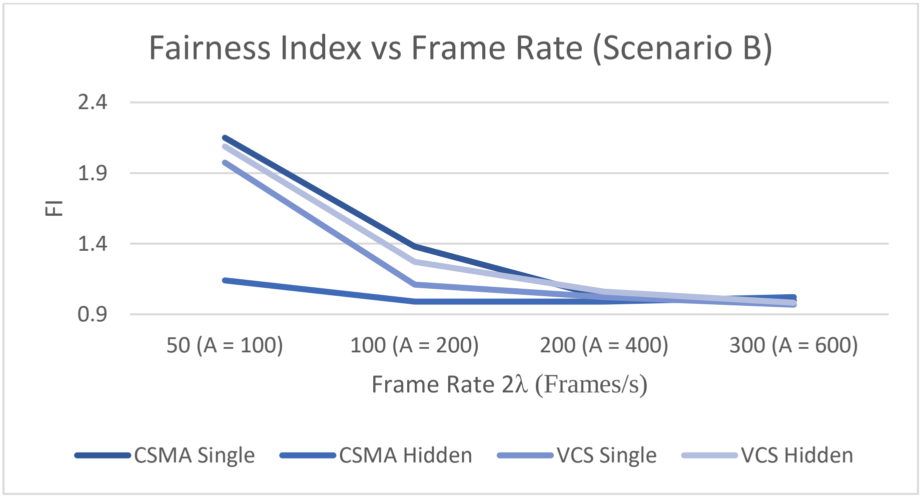


Figure 10: *FI vs Frame Rate for all types, frame rate scenario B*

In this scenario, A’s frame rate doubles C’s at any time. This is extremely noticeable when station A has frame rate 100fps and station C has frame rate 50ps. At this point, the total traffic hasn’t reached saturation, so A is able to make full use of the double frame rate with FI about 2 for each protocol/domain combination except for CSMA Hidden. CSMA hidden does not see the same increase in FI as the higher frame rate and blind station communication results in too many packet collisions. As the frame rates increase, the throughput begins to saturate, and thus the fairness increases approaches unity.