ECE 578 Project1. DCF of 802.11

Tianrui Tan, Jimmy Tran

November 1, 2023

1 Introduction

1.1 Transmission Topology

In this project, our objective is to investigate the performance of various access protocols within a wireless environment, with a specific focus on two scenarios illustrated in Figure 1. 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.

1.2 Responsibilities

The tasks among the team were split as follows: Both members participated in the programming and simulation, with Tianrui's idea of coding the problem as a state machine, and Jimmy addressing bugs and validating the event-based simulation. Tianrui developed the base logic for how the stations should function, implementing the various phases of a contention period through a state diagram, in addition to other station-related logic such as contention windows and Poisson-generated arrivals. Using this station logic, Jimmy implemented the logic at the access point that assesses when collisions have happened, when the access point is transmitting, and keeps track of time slots throughout the simulation. Once the simulation logic for DCF-topology(a) was completed, Jimmy extended Tianrui's state machine to accommodate for states related to VCS, and both members jointly worked together to extend the code for DCF-topology(a) to apply to other topology as well. After achieving a satisfactory level of simulation integrity, Tianrui created the outline and rough draft for the report, while Jimmy addressed any remaining bugs and odd scenarios in the simulation. Subsequently, Jimmy conducted peer reviews and edits of the report, while Tianrui generated graphs based on the generated simulation data. Finally, Tianrui and Jimmy collaboratively analyzed the data generated from the simulation.

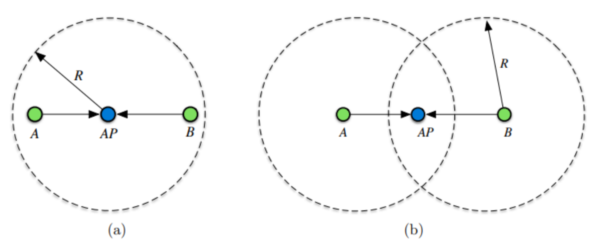


Figure 1: (a) The configuration for concurrent transmissions occurring in a shared collision domain, (b) the arrangement for concurrent transmissions when A and B function as hidden terminals.

2 Simulation and Development

All the programming is done in Python. In our simulation, we integrated the concept of a state machine. As shown in Figure 3, both the stations and the access point transition through different states in response to specific events and scenarios outlined in the problem statement. The terminals then use the feedback received to determine their next state. The simulation results are shown in Figure 2.

At the start of any given simulation, each station uses the Poisson distribution to generate an array indicating the slots at which frames will arrive. Starting from the first given arrival slot in this array, each station keeps track of a counter that will increment through this array after each successful transmission. If the current time slot in the simulation is equal to or exceeds this counter, the station will always have at least one frame to try to transmit to the access point at the start of each contention period. If the next arrival slot exceeds the current simulation slot, the station remains in the free state until the slot when the arrival should happen.

When a station attempts to transmit a frame, it switches to the sensing (DIFS) state, followed by the backoff state, and waits an appropriate number of slots according to the DCF 802.11 specifications. After completing backoff, if VCS is enabled, the station enters a unique cycle of states for RTS and CTS, as seen in Figure 3b; otherwise, the station immediately enters the sending state. In topology(a), if another station starts transmitting, the station will switch from the sensing/backoff state into the waiting for NAV state until the transmission is complete, retaining its backoff value if it was already in the backoff state. In topology(b), stations will not enter the waiting for NAV state unless VCS is enabled, in which case a station will enter the waiting for NAV state when the access point tries to send a CTS to another station. Rare cases where one station starts transmitting when the access point enters transmit mode, such as during an ACK or a CTS, are handled with special logic in the simulation to reproduce those unique station behaviors. All of our simulation and development can be accessed HERE!

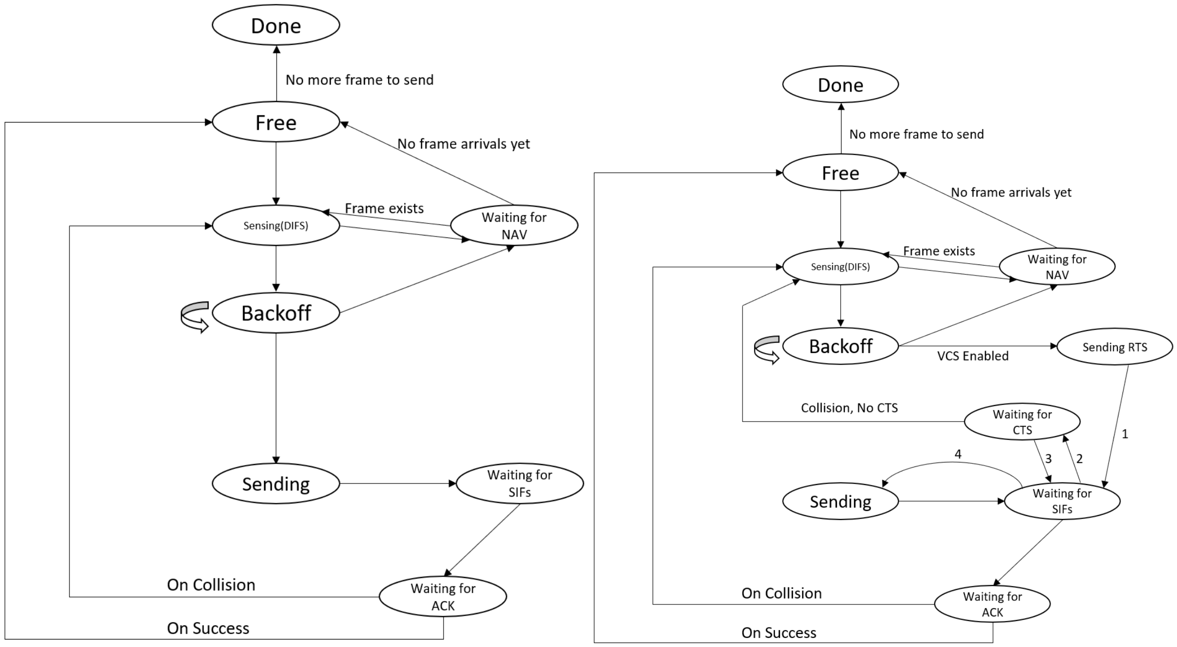
The simulation's parameters are detailed in the Table 1. To calculate the data frame size in slots, $\big((1500 \text{ bytes} \times 8) \ / \ 10\text{Mpbs}\ \big) \ / \ 10\ \mu\text{s} = 120$ slots.

		Frame Rate (frames / sec)					
		100	**200**	**300**	**500**	**800**	**1000**
DCF-topology(a)	**Successes (A)**	1000	2002	3003	3431	3542	3450
	Successes (B)	1001	1999	3000	3472	3327	3428
	Collisions (AP)	1	8	18	758	791	782
DCF-topology(b)	**Successes (A)**	993	1204	1535	1465	1740	1116
	Successes (B)	999	1424	1211	1270	1066	1670
	Collisions (A)	1412	3296	3410	3354	3518	3154
	Collisions (B)	1413	3414	3236	3262	3151	3443
DCF/VCS-topology(a)	**Successes (A)**	999	1997	2997	3507	3650	3681
	Successes (B)	1000	1998	3001	3624	3486	3456
	Collisions (AP)	2	5	7	828	777	809
DCF/VCS-topology(b)	**Successes (A)**	999	1998	3002	4696	5217	4181
	Successes (B)	999	1999	3001	2383	1891	2929
	Collisions (A)	22	73	1469	359	303	274
	Collisions (B)	27	64	1432	343	299	276

Figure 2: The simulation results

Parameter	Value	Parameter	Value
Data frame size	1,500 bytes	ACK, RTS, CTS size	3 slots
Slot duration	10 µs	DIFS duration	4 slots
SIFS duration	1 slot	Bandwidth	10 Mbps
CW_0	8 slots	CW_{max}	512 slots
$\lambda_c = \lambda_A$	{100, 200, 300, 500, 800, 1000} frames/sec	Simulation time	10 sec

Table 1: Simulation parameters.



(a) VCS disabled (b) VCS enabled

Figure 3: State diagrams

3 Analysis

3.1 Throughput

The formulation used for our analysis of throughput is shown in Equation 1. Figures 4a and 4b show the throughput of stations A and B as a function of the arrival rate of each station, respectively.

$$T = \frac{\text{data frame size} \times 8 \times \text{successful transmission}}{\text{simulation time}} \tag{1}$$

3.2 Collisions

Collisions are detected by assessing whether the node states are concurrently in the 'transmitting' state. The Collision vs. λ graphs are represented in Figure 5.

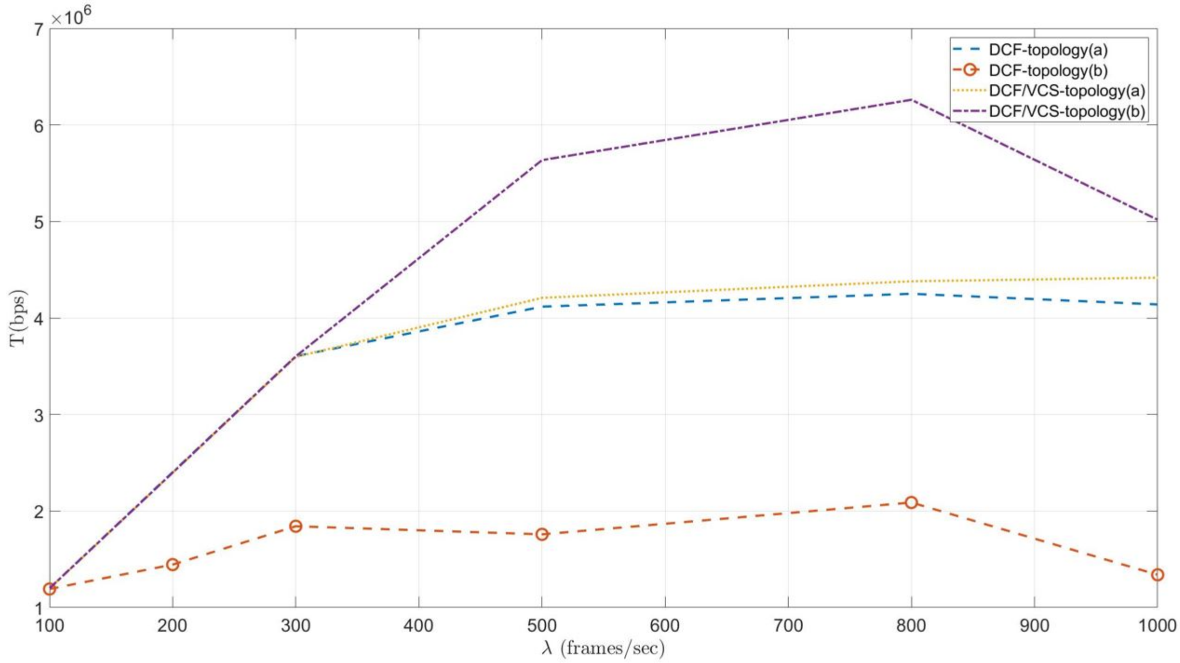
3.3 Fairness Index

In the field of network engineering, fairness metrics play a crucial role in assessing whether system resources are distributed equitably among users or applications. Various mathematical and conceptual definitions of fairness exist for this purpose. In this report we will use a simple fairness index metric, Equation 2. If the FI value exceeds 1, it indicates a higher level of fairness towards station A, and conversely, as shown in Figure 6,

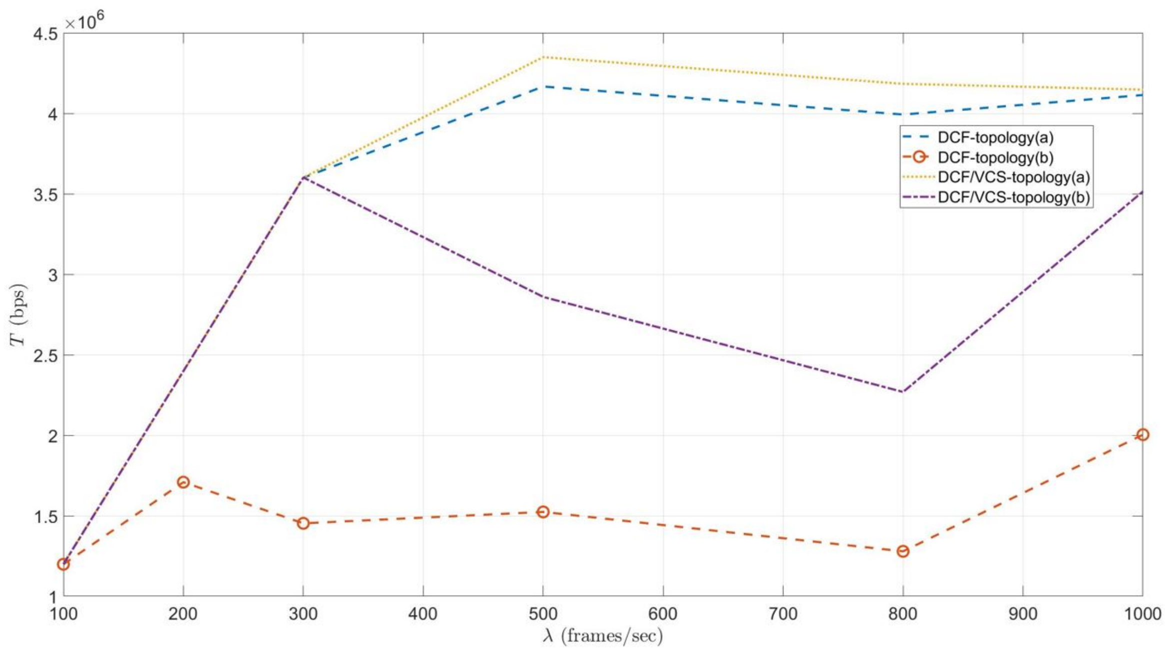
$$FI = \frac{Success_A + Collision_A}{Success_B + Collision_B} \tag{2}$$

3.4 Conclusion

Initially, for arrival rates of $\lambda \leq 300$ frames/sec, note that the throughput of both stations in Figures 4a and 4b scales linearly with the arrival rate for two of the topologies— DCF-topology(a), DCF/VCS-topology(a)— while the number of collisions is almost zero. This is due to the lower arrival rate, which means that the arrivals of A and B are sparser, thus reducing the probability of contention. This aligns



(a) Station A throughput

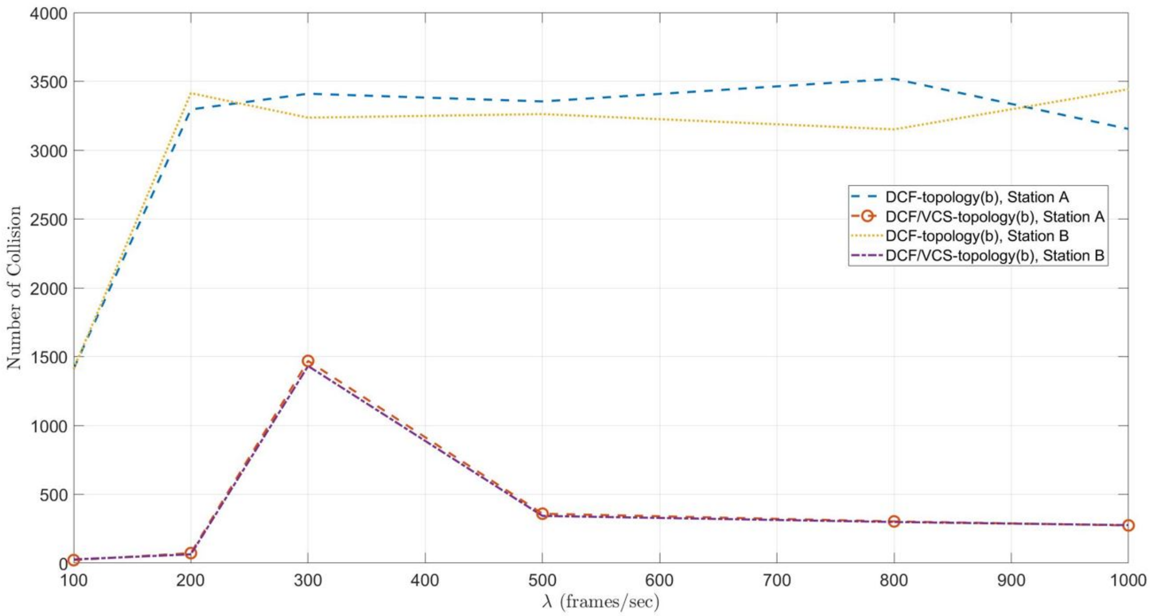


(b) Station B throughput

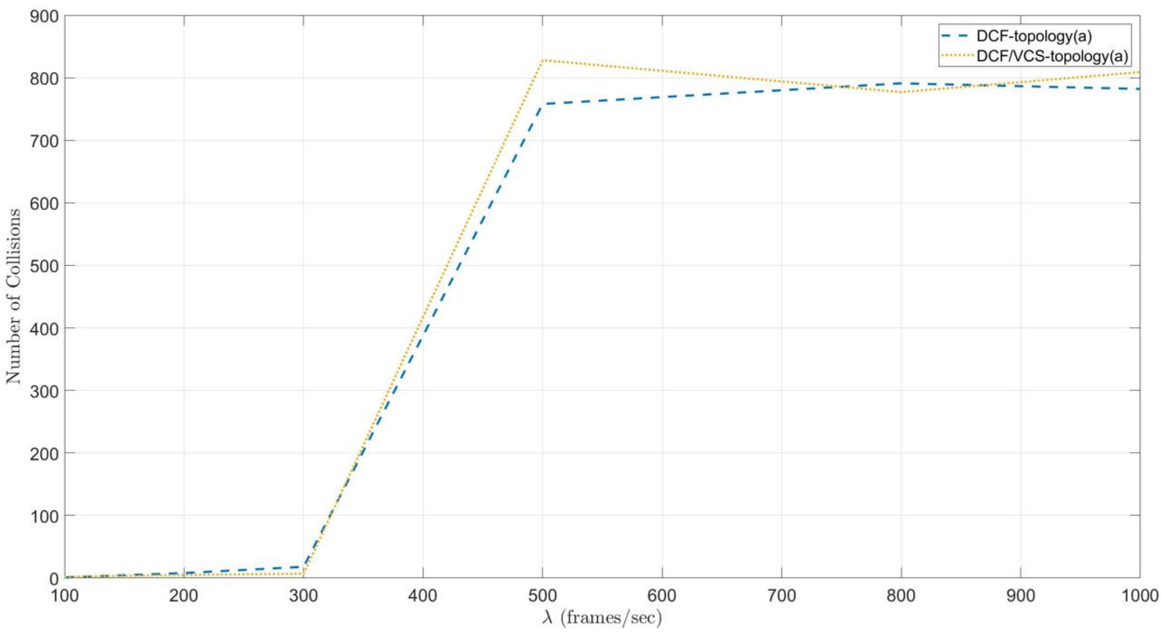
Figure 4: Throughput T

with what we see in Figure 5b as both stations experience less collisions at lower arrival rates. Because the number of collisions is close to zero for both shared collision domain topology at arrival rates $\lambda \leq 300$ frames/sec, we also see that the throughput matches the arrival rate. In other words, for arrival rates of 100, 200, and 300 frames/sec, we see that the throughput is close to $100 \times 12000/10$ bits/sec, $200 \times 12000/10$ bits/sec, and $300 \times 12000/10$ bits/sec.

At arrival rates of $\lambda \geq 500$ frames/sec, we start to see more collisions in figure 5(b) for both shared collision domain topology, as the higher arrival rates start to create a backlog of frames that need to be sent for each station. This means that both station A and station B spend more time contending for use of the channel at the same time, increasing the probability of collision. This increase in collisions,



(a) Stations collision in hidden terminal scenario



(b) AP collisions in shared domain

Figure 5: Number of Collisions

		Frame Rate (frames / sec)					
		100	200	300	500	800	1000
DCF-topology(a)	**Fairness Index**	0.999002	1.001495	1.000994	0.990307	1.05221	1.005226
DCF-topology(b)	**Fairness Index**	0.997098	0.930136	1.111986	1.063327	1.246858	0.835126
DCF/VCS-topology(a)	**Fairness Index**	0.999002	0.999501	0.99867	0.97372	1.038471	1.052755
DCF/VCS-topology(b)	**Fairness Index**	0.995127	1.003878	1.008572	1.854365	2.520548	1.390016

Figure 6: The calculated FI table

in turn, reduces the throughput for both topology. However, we do see a slightly higher throughput for both stations in the shared collision domain topology when VCS is enabled compared to the topology

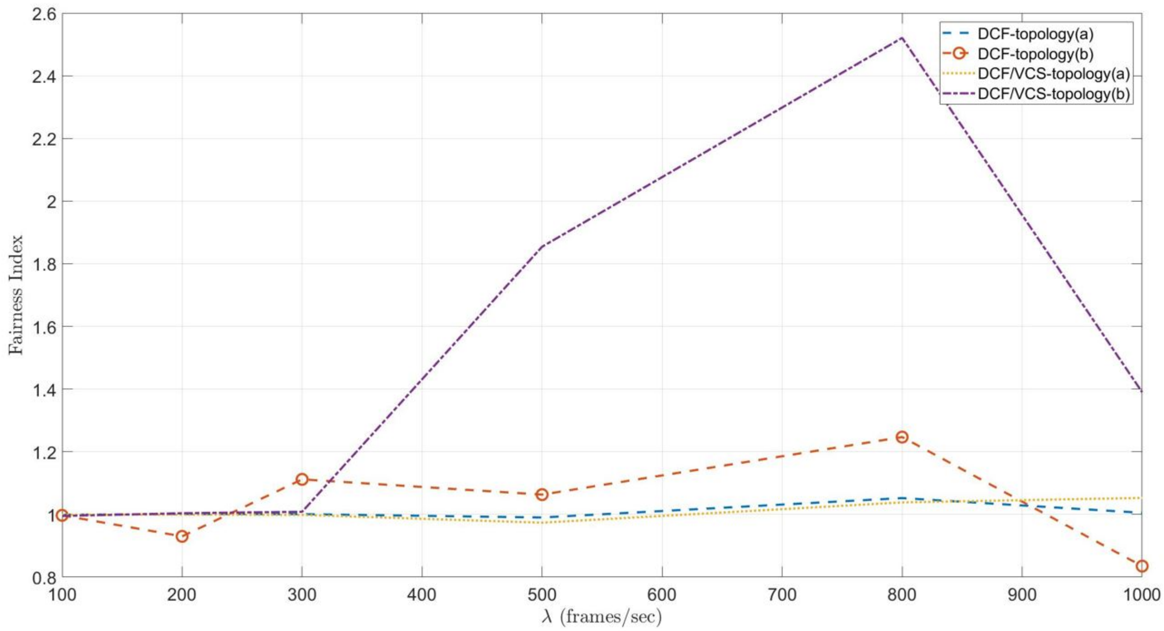


Figure 7: The Fairness Index

where VCS is not enabled, as seen in Figures 4a and 4b. This increase in throughput is the result of one of the benefits of VCS, because the only collisions that occur between stations in a shared collision domain with VCS are collisions involving RTS frames. RTS frames are much smaller than the actual data frames being sent, so both stations spend less time colliding with each other and more time trying to transmit, thus increasing throughput.

With DCF-topology(b), stations A and B are unable to listen to the other station's transmissions, which means that a collision is likely to happen at any time during another station's transmission, greatly increasing the probability of collision. Even at the lowest arrival rate of $\lambda = 100$ frames/sec, we see in Figures 5a and 5b that the number of collisions for DCF-topology(b) is much greater than the number of collisions for other topology. This increase in collisions also results in throughput that is much less compared to other topology, as seen in Figures 4a and 4b, as both stations spend more time colliding and more time backing off due to constantly having to increase contention windows after collision, resulting in less time spent on successful transmissions. From Figure 7, we also see that the fairness index of DCF-topology(b) is less stable and more skewed towards one station compared to the shared collision domain topology, especially at higher arrival rates. This is because when one station wins a contention period after multiple consecutive collisions, it resets its contention window, allowing the winning station to repeatedly seize the channel, as the other station remains at the higher contention window. On a separate note, note that the number of collisions plateauing at arrival rates $\lambda \geq 200$ frames/sec is due to the properties of the channel, the frame size, as well as the simulation time, and does not imply anything about the likelihood of collisions at higher arrival rates.

For analysis of DCF/VCS-topology(b), consider the case where both stations have a backlog of frames to send and immediately try to send their frames once the channel is free (this is almost always the case at higher arrival rates). Because neither station can hear the others' RTS, and because an RTS takes up three slots and an SIFS takes up one slot, for a station to seize the channel, the station must pick a backoff value that is at least five greater than the backoff value of the other station. In the best case scenario where each station has the minimum contention window size of eight, each station has a 6/64, or 9.375% chance to win the contention period, meaning that there is a high probability of 81.25% for a collision to occur. Using this information and the trends that we see from the figures, we can conclude the following:

- In Figure 5a, the spike in the number of collisions at arrival rate $\lambda = 300$ frames/sec is due to the high probability of collisions when both stations have smaller contention windows. At a moderate arrival rate, the arrival rate of frames is high enough that both stations will spend more time simultaneously contending for the channel, but also low enough that the arrivals of frames are still somewhat sparse, allowing for each station to have a chance to capture the channel before another contention period starts. This means that the contention window for each station resets their contention window fairly often, which increases the probability of collisions.

Note: However, despite the increased number of collisions, we still see in Figures 4a and 4b that the throughput of DCF/VCS-topology(b) scales fairly linearly with respect to arrival rate without any reduction in throughput, for $\lambda \leq 300$ frames/sec. Once again, this is due to VCS reducing the amount of time spent on collisions by restricting collisions to RTS, which increases time spent transmitting.

- At arrival rates of $\lambda \geq 500$ frames/sec, we start to see the number of collisions gradually decline for both stations in Figure 5a, and at the same time we also see that one station has much higher throughput than the other station in Figures 4a and 4b. This happens because when one station wins a contention period, it lowers its contention window back to the minimum, while the other station remains at its higher contention window. Due to the already high probability of collisions for this topology even at the minimum contention window, it becomes more difficult for the station that lost the previous contention window to win subsequent contention periods, allowing the station that won previously to continue seizing the channel and preventing the losing channel from resetting its contention window. This is further verified by looking at the how the fairness index of DCF/VCS-topology(b) in Figure 7 tends to skew away from 1 at higher arrival rates.