Andrew Camps

Derek Paris

October 10, 2017

ECE 478 Project 1

Loukas Lazos

**The Distributed Coordination Function of 802.11**

As a team Derek Paris and Andrew Camps studied the performance of multiple access protocols in a wireless setting. The project studied two different topological orderings of nodes, two different setups of Carrier Sensing Multiple Access (CSMA), all at different packet rate distributions. To complete this project as a team we decided to meet up on multiple occasions and work on creating a simulation for each test. We worked together to list out the many possible edge cases, then design a simulation which would deal with each case appropriately. After working out much of the needed logic, Andrew began to code the simulations for the first few scenarios. Derek and Andrew peer programmed many of the more logically intense edge cases. Derek code reviewed the rest of the code periodically as it was written.

In order to test all of the different scenarios in the project our team decided that we would develop this simulation is C/C++. The simulation consisted of using a single main.cpp file which, when compiled and run, outputted to the terminal the necessary data to compile the graphs in the figures below. Each time the program was run, the simulation was conducted 1000 times for each of the many scenarios. The data retrieved from each of these trials was then averaged and outputted. This data included: throughput averages, fairness index, collision average, total slots used averages, and success averages. The program was broken down into 4 major functions, where both topologies, node in either the same or different collision domains, each used CSMA and CSMA (with VCS) functions. Each of these 4 functions was run at different transmission rates. These rates we tested in two different sets. The first transmission rate set was a set of 50, 100, 200, and 300 pkts/sec for both nodes A and C. The second set doubled the transmission rates of node A, but left the rates for node C unchanged. See below for the graphical representation of each simulation.

Figure 1: Average Throughput of Node A (λA = λC)

Figure 1 shows average throughput of node A, while λA = λC, throughout the 1000 simulation tests. What we found was that in Scenario A, all nodes in the same collision domain, the type of CSMA (VCS or non-VCS) did not substantially affect the throughput of Node A. This is shown in the solid gray and dashed yellow line in Figure 1. In Scenario B, however, for only rates greater than 100 Mbps, CMSA with VCS far outperformed CSMA without VCS. At a transmission rate of 200 Mbps, CSMA-VCS outperforms CSMA by 33.7%. At 300 Mbps, that number is 45.7%. These can be seen in the solid blue and dashed orange line in Figure 1. At transmission rates at or less than 100 Mbps, there is no significant difference in throughput for any scenario with any type of CSMA. What this suggests to us is that at lower transmission rate, efficient throughput is much more easily achieved and requires less optimization.

Figure 2: Average Throughput of Node A (λA = 2λC)

Figure 2 shows average throughput of node A, while λA = 2λC, throughout the 1000 simulation tests. What we found was that in Scenario A, all nodes in the same collision domain, the type of CSMA (VCS or non-VCS) did not substantially affect the throughput of Node A, though the difference between the two did show more than in the λA = λC case. This is shown in the solid blue and dashed orange line in Figure 2. In Scenario B, however, for only rates greater than 100 Mbps, CMSA with VCS far outperformed CSMA without VCS. At a transmission rate of 200 Mbps, CSMA-VCS outperforms CSMA by 51.7%. At 300 Mbps, that number is 87.1%. These can be seen in the solid gray and dashed yellow line in Figure 2. It is interesting to note that even though the transmission rate of node A was doubled, we do not see a divergence in throughput values until the after the 2nd data point, 200 Mbps. In Figure 1, we also did not see the divergence until after the 2nd data point, however it was only 100 Mbps. Further research may be required to discover why this is so. Also note that in throughput in Scenario B with CSMA-VCS far outperformed every other situation. This will be discussed further after Figure 4, where throughput of node C is also analyzed.

Figure 3: Average Throughput of Node C (λA = λC)

Figure 3 shows average throughput of node C, while λA = λC, throughout the 1000 simulation tests. What we found was that in Scenario A, all nodes in the same collision domain, the type of CSMA (VCS or non-VCS) did not substantially affect the throughput of Node C. That is, until we surpassed a transmission rate of 200 Mbps. Then there is a slight divergence between CSMA and CSMA-VCS. This is shown in the solid blue and dashed orange line in Figure 3. In Scenario B, however, for only rates greater than 100 Mbps, CMSA with VCS far outperformed CSMA without VCS. At a transmission rate of 200 Mbps, CSMA-VCS outperforms CSMA by 30.9%. At 300 Mbps, that number is 41.3%. These can be seen in the solid gray and dashed yellow line in Figure 3.

Figure 4: Average Throughput of Node C (λA = 2λC)

Figure 4 shows average throughput of node c, while λA = 2λC, throughout the 1000 simulation tests. What we found was that in Scenario A, all nodes in the same collision domain, the type of CSMA (VCS or non-VCS) did not substantially affect the throughput of Node C. This is shown in the solid blue and dashed orange line in Figure 2. In Scenario B, however, for only rates greater than 100 Mbps but also less than 250 Mbps, CSMA with VCS outperforms CSMA without VCS. At transmission rates greater than 250 Mbps, throughput with CSMA-VCS drops below CSMA without VCS. This can be seen in the solid gray and dashed yellow line in Figure 2. This drop in throughput in Node C is likely connected to the spike in node A’s throughput that we observed in Figure 2. What probably happened, is that at higher transmission values, node A “blocks out” node C. This occurs when there are so many collisions and each node’s backoff time grows so large, that as soon as one node transmits successfully and resets its backoff time, it can rapidly transmit packets while the other node is stuck backing off.

Figure 5: Average Collisions of Node A (λA = λC)

Figure 6: Average Collisions of Node A (λA = 2λC)

Figure 7: Average Collisions of Node C (λA = λC)

Figure 8: Average Collisions of Node C (λA = 2λC)

Figure 9: Fairness Index (λA = λC)

Figure 10: Fairness Index (λA = 2λC)