ENSC 427: COMMUNICATION NETWORKS
SPRING 2025

Performance analysis of 802.11ax 2.4GHz (WiFi 6)
Final Project: Team 10

http://sfu.ca/~kcl35/ensc427

Owen Coukell	301426374	orc@sfu.ca
Kurtis Lew	301384672	kurtisl@sfu.ca
Katherine Lee	301265784	kcl35@sfu.ca

1. Abstract

Following the release of IEEE 802.11ac (WiFi 5) in 2013, IEEE 802.11ax (WiFi 6) was released in 2019 [1]. A performance analysis of the 802.11ax 2.4GHz (WiFi 6) protocol lists multiple benefits, with a major highlight being the inclusion of an improved MU-MIMO (Multi-User Multiple-Input Multiple-Output) [2] which allows for simultaneous downlink and uplink. This increases the maximum number of simultaneous streams from four to eight [3], boosting maximum speeds to yield impressive throughput. This project will outline the main characteristics of the WiFi 6 protocol, indicating if there are any significant changes compared to previous iterations and displaying the outcomes of key variable manipulation in our code simulations.

2. Introduction

Just over two decades after the initial release of Wi-Fi, consistent development and improvement led to the release of Wi-Fi 6, also known as IEEE 802.11ax. These improvements include, but are not limited to, an increased number of possible simultaneous users per host, appreciable speed boosts, enhanced security, and lowered power consumption. To better understand the Wi-Fi 6 protocol's performance on a quantitative level, we simulated a Wi-Fi 6 network through ns-3 and measured its performance in terms of latency and throughput for clients running different types of payloads.

As the number of connected devices continues to grow, Wi-Fi networks face increasing demands in maintaining low latency, high reliability, and efficient data transmission, especially in dense environments. Wi-Fi 6 introduces several advancements to address these issues, promising significant improvements in latency and reliability compared to previous generations. These enhancements can be attributed to the use of Orthogonal Frequency Division Multiple Access (OFDMA) [4] and Basic Service Set (BSS) colouring, which improve network efficiency and reduce interference [5], alongside an upgrade to Multiple-User Multiple-Input Multiple-Output (MU-MIMO), which now allows for both uplink and downlink of multiple devices at the same time [3].

Through these improvements, Wi-Fi 6 boasts much greater potential for overall throughput whilst managing more devices than previously possible. Understanding how Wi-Fi 6 performs when various numbers of streams, data transfer speeds, and other restrictions are placed on a network allows for a simple yet accurate simulation of a real-world Wi-Fi performance, as most users are not capable of affording the maximum potential of any Wi-Fi version. It is important to note that all simulations in this report are done with an isolated Wi-Fi network and that additional factors such as device/wire quality and network interference would occur in more accurate real-world situations. Aside from examining these issues, much can be gleaned from an analysis of latency and throughput in a single network simulation. As such, this study seeks to evaluate Wi-Fi 6 in a vacuum to showcase major improvements and output statistics without the risk of real-world interference. The main elements that will be analyzed here are the channel widths and the modulation coding scheme, as they increase in proportion to the throughput with some interesting variations.

Related Works

I. Performance and Simulation Analysis of 802.11ax OFDMA in Contention-driven Scenarios

   This paper validated the implementation of OFDMA in ns-3 simulation results against theoretical models, aiming to demonstrate that within contention-driven use cases, the simulation-based performance would be supportive of the analytical predictions. This paper also highlighted the efficiency in improving throughput and reducing latency in congested environments. [6]

II. Implementation and Evaluation of IEEE 802.11ax Channel Sounding Frame Exchange in ns-3

   The researchers implemented and evaluated IEEE 802.11ax's MU-MIMO channel sounding in ns-3, which enabled a realistic simulation of beamforming and spatial multiplexing. MU-MIMO allows multiple users to transmit and receive data at the same time, a key feature of Wi-Fi 6. This helps improve performance in multi-device scenarios, paving the way to properly evaluating the impact of channel sounding on network performance. [7]

III. Comparative Performance Analysis of the IEEE 802.11ax and 802.11ac MIMO Link For WLANs

   In this analysis, ns-3 was used to measure throughput performance and maximum achievable distance, comparing Wi-Fi 6 and Wi-Fi 5. They showed that both Wi-Fi 5 and Wi-Fi 6 had significant gains, attributing these enhancements to improvements in the MIMO, OFDMA, and modulation schemes. Overall, they highlighted that Wi-Fi 6 showed even more improvements than Wi-Fi 5. [8]

IV. An Optimization of Network Performance in IEEE 802.11ax Dense Networks

   In this paper, the researchers investigated network performance for optimized IEEE802.11ax dense networks using ns3 over 3 different dense network topologies. Highlight the importance of CCA Threshold, transmitter power, antenna gain, and receiver sensitivity to improving efficiency in dense networks, and how detrimental frame collisions are to network efficiency. [9]

V. Wi-Fi 6: Performance Analysis of IEEE 802.11AX

   This paper focused on evaluating the performance of Wi-Fi 6 on draft hardware against both synthetic and real-world benchmarks. Over a variety of environments, both indoor and outdoor, they demonstrated how much better Wi-Fi 6 was in both ideal and challenging environments. They highlighted the significant improvements in throughput, power efficiency, and range of Wi-Fi 6 compared to previous generations. [10]

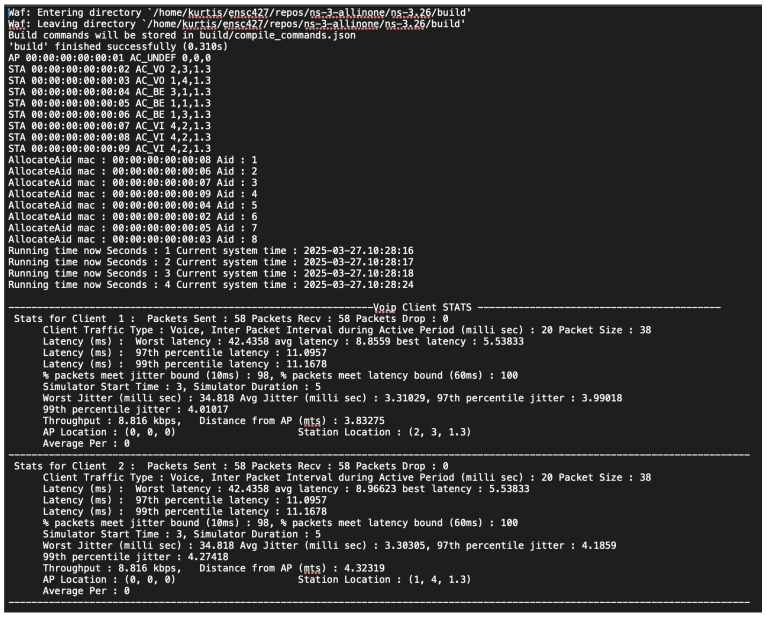
3. Main Section

All testing in this project was conducted using the Cisco ns-3 Wi-Fi 6 simulation tool posted publicly on Github [11]. This allowed us to create a controlled environment to analyze how a Wi-Fi 6 network would perform under different workloads. While recent versions of ns-3 can simulate networks based on older Wi-Fi versions via the newest built-in modules, it was presumed early on that the Cisco codebase would guarantee that we could simulate a Wi-Fi 6 network with a reasonable amount of customization.

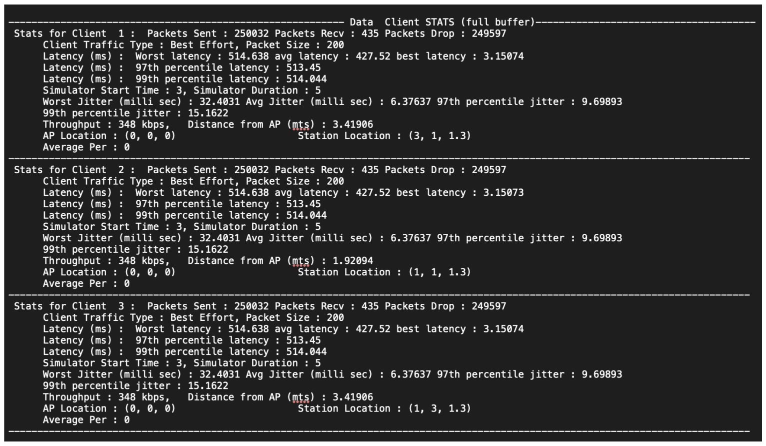
In short, the simulation setup procedure was as follows:

- A clean install of ns-3.26 was installed on a Debian 11 virtual machine
- The Cisco custom modules were applied to the clean install of ns-3.26
- Compiling the sample code provided a simulated network with one Wi-Fi 6 2.4GHz access point and eight clients demanding various payloads
    - All nodes, including both access points and clients, could be customized with unique MAC addresses and positions
    - As per the default simulation scenario, the clients were placed roughly equidistant from the access point, as not to let distance from the access point affect performance metrics
    - The client types were as follows:
        - 2x VoIP Clients
        - 3x Data Clients
        - 3x Video Clients

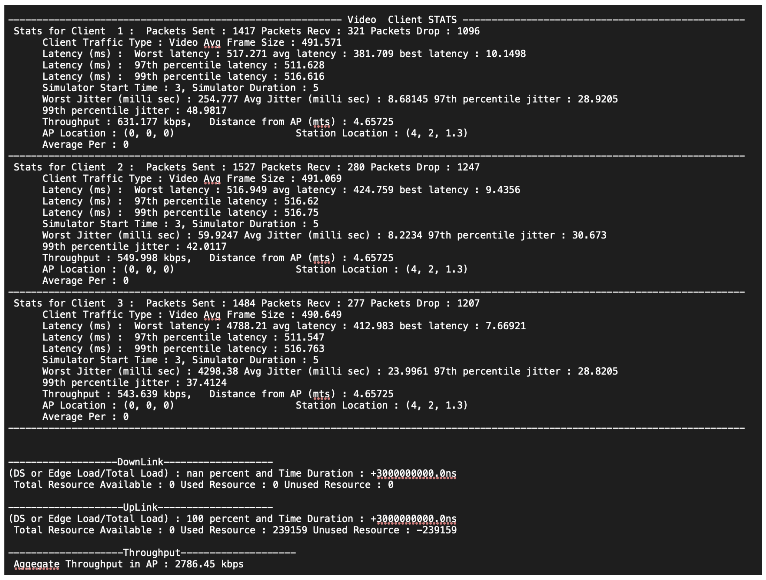
A sample of the logging information provided by the simulation during runtime is provided below in Figs 1a-1c.



```
Waf: Entering directory `/home/kurtis/ensc427/repos/ns-3-allinone/ns-3.26/build'
Waf: Leaving directory `/home/kurtis/ensc427/repos/ns-3-allinone/ns-3.26/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.310s)
AP 00:00:00:00:00:01 AC_UNDEF 0,0,0
STA 00:00:00:00:00:02 AC_VO 2,3,1.3
STA 00:00:00:00:00:03 AC_VO 1,4,1.3
STA 00:00:00:00:00:04 AC_BE 3,1,1.3
STA 00:00:00:00:00:05 AC_BE 1,1,1.3
STA 00:00:00:00:00:06 AC_BE 1,3,1.3
STA 00:00:00:00:00:07 AC_VI 4,2,1.3
STA 00:00:00:00:00:08 AC_VI 4,2,1.3
STA 00:00:00:00:00:09 AC_VI 4,2,1.3
AllocateAid mac : 00:00:00:00:00:08 Aid : 1
AllocateAid mac : 00:00:00:00:00:06 Aid : 2
AllocateAid mac : 00:00:00:00:00:07 Aid : 3
AllocateAid mac : 00:00:00:00:00:09 Aid : 4
AllocateAid mac : 00:00:00:00:00:04 Aid : 5
AllocateAid mac : 00:00:00:00:00:02 Aid : 6
AllocateAid mac : 00:00:00:00:00:05 Aid : 7
AllocateAid mac : 00:00:00:00:00:03 Aid : 8
Running time now Seconds : 1 Current system time : 2025-03-27.10:28:16
Running time now Seconds : 2 Current system time : 2025-03-27.10:28:17
Running time now Seconds : 3 Current system time : 2025-03-27.10:28:18
Running time now Seconds : 4 Current system time : 2025-03-27.10:28:24

---------------------------------------------------------------Voip Client STATS ------------------------------------------
 Stats for Client  1 :  Packets Sent : 58 Packets Recv : 58 Packets Drop : 0
      Client Traffic Type : Voice, Inter Packet Interval during Active Period (milli sec) : 20 Packet Size : 38
      Latency (ms) :  Worst latency : 42.4358 avg latency : 8.8559 best latency : 5.53833
      Latency (ms) :  97th percentile latency : 11.0957
      Latency (ms) :  99th percentile latency : 11.1678
      % packets meet jitter bound (10ms) : 98, % packets meet latency bound (60ms) : 100
      Simulator Start Time : 3, Simulator Duration : 5
      Worst Jitter (milli sec) : 34.818 Avg Jitter (milli sec) : 3.31029, 97th percentile jitter : 3.99018
      99th percentile jitter : 4.01017
      Throughput : 8.816 kbps,   Distance from AP (mts) : 3.83275
      AP Location : (0, 0, 0)                     Station Location : (2, 3, 1.3)
      Average Per : 0
-------------------------------------------------------------------------------------------------------------------------------
 Stats for Client  2 :  Packets Sent : 58 Packets Recv : 58 Packets Drop : 0
      Client Traffic Type : Voice, Inter Packet Interval during Active Period (milli sec) : 20 Packet Size : 38
      Latency (ms) :  Worst latency : 42.4358 avg latency : 8.96623 best latency : 5.53833
      Latency (ms) :  97th percentile latency : 11.0957
      Latency (ms) :  99th percentile latency : 11.1678
      % packets meet jitter bound (10ms) : 98, % packets meet latency bound (60ms) : 100
      Simulator Start Time : 3, Simulator Duration : 5
      Worst Jitter (milli sec) : 34.818 Avg Jitter (milli sec) : 3.30305, 97th percentile jitter : 4.1859
      99th percentile jitter : 4.27418
      Throughput : 8.816 kbps,   Distance from AP (mts) : 4.32319
      AP Location : (0, 0, 0)                     Station Location : (1, 4, 1.3)
      Average Per : 0
-------------------------------------------------------------------------------------------------------------------------------
```



```
---------------------------------------------------------- Data  Client STATS (full buffer)----------------------------------------
 Stats for Client  1 :  Packets Sent : 250032 Packets Recv : 435 Packets Drop : 249597
      Client Traffic Type : Best Effort, Packet Size : 200
      Latency (ms) :  Worst latency : 514.638 avg latency : 427.52 best latency : 3.15074
      Latency (ms) :  97th percentile latency : 513.45
      Latency (ms) :  99th percentile latency : 514.044
      Simulator Start Time : 3, Simulator Duration : 5
      Worst Jitter (milli sec) : 32.4031 Avg Jitter (milli sec) : 6.37637 97th percentile jitter : 9.69893
      99th percentile jitter : 15.1622
      Throughput : 348 kbps,   Distance from AP (mts) : 3.41906
      AP Location : (0, 0, 0)                     Station Location : (3, 1, 1.3)
      Average Per : 0
-------------------------------------------------------------------------------------------------------------------------------
 Stats for Client  2 :  Packets Sent : 250032 Packets Recv : 435 Packets Drop : 249597
      Client Traffic Type : Best Effort, Packet Size : 200
      Latency (ms) :  Worst latency : 514.638 avg latency : 427.52 best latency : 3.15073
      Latency (ms) :  97th percentile latency : 513.45
      Latency (ms) :  99th percentile latency : 514.044
      Simulator Start Time : 3, Simulator Duration : 5
      Worst Jitter (milli sec) : 32.4031 Avg Jitter (milli sec) : 6.37637 97th percentile jitter : 9.69893
      99th percentile jitter : 15.1622
      Throughput : 348 kbps,   Distance from AP (mts) : 1.92094
      AP Location : (0, 0, 0)                     Station Location : (1, 1, 1.3)
      Average Per : 0
-------------------------------------------------------------------------------------------------------------------------------
 Stats for Client  3 :  Packets Sent : 250032 Packets Recv : 435 Packets Drop : 249597
      Client Traffic Type : Best Effort, Packet Size : 200
      Latency (ms) :  Worst latency : 514.638 avg latency : 427.52 best latency : 3.15074
      Latency (ms) :  97th percentile latency : 513.45
      Latency (ms) :  99th percentile latency : 514.044
      Simulator Start Time : 3, Simulator Duration : 5
      Worst Jitter (milli sec) : 32.4031 Avg Jitter (milli sec) : 6.37637 97th percentile jitter : 9.69893
      99th percentile jitter : 15.1622
      Throughput : 348 kbps,   Distance from AP (mts) : 3.41906
      AP Location : (0, 0, 0)                     Station Location : (1, 3, 1.3)
      Average Per : 0
-------------------------------------------------------------------------------------------------------------------------------
```

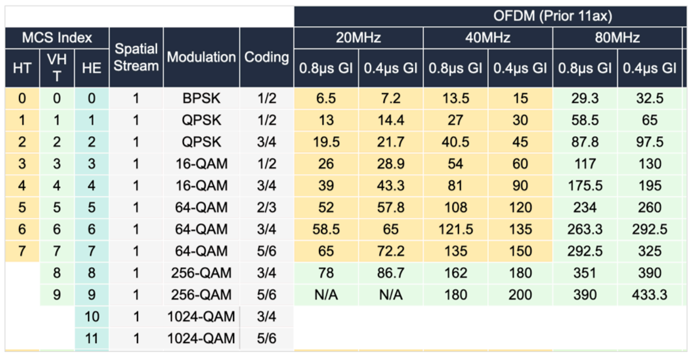


```
---------------------------------------------------------- Video  Client STATS ------------------------------------------------
 Stats for Client  1 :  Packets Sent : 1417 Packets Recv : 321 Packets Drop : 1096
      Client Traffic Type : Video Avg Frame Size : 491.571
      Latency (ms) :  Worst latency : 517.271 avg latency : 381.709 best latency : 10.1498
      Latency (ms) :  97th percentile latency : 511.628
      Latency (ms) :  99th percentile latency : 516.616
      Simulator Start Time : 3, Simulator Duration : 5
      Worst Jitter (milli sec) : 254.777 Avg Jitter (milli sec) : 8.68145 97th percentile jitter : 28.9205
      99th percentile jitter : 48.9817
      Throughput : 631.177 kbps,   Distance from AP (mts) : 4.65725
      AP Location : (0, 0, 0)                     Station Location : (4, 2, 1.3)
      Average Per : 0
-------------------------------------------------------------------------------------------------------------------------------
 Stats for Client  2 :  Packets Sent : 1527 Packets Recv : 280 Packets Drop : 1247
      Client Traffic Type : Video Avg Frame Size : 491.069
      Latency (ms) :  Worst latency : 516.949 avg latency : 424.759 best latency : 9.4356
      Latency (ms) :  97th percentile latency : 516.62
      Latency (ms) :  99th percentile latency : 516.75
      Simulator Start Time : 3, Simulator Duration : 5
      Worst Jitter (milli sec) : 59.9247 Avg Jitter (milli sec) : 8.2234 97th percentile jitter : 30.673
      99th percentile jitter : 42.0117
      Throughput : 549.998 kbps,   Distance from AP (mts) : 4.65725
      AP Location : (0, 0, 0)                     Station Location : (4, 2, 1.3)
      Average Per : 0
-------------------------------------------------------------------------------------------------------------------------------
 Stats for Client  3 :  Packets Sent : 1484 Packets Recv : 277 Packets Drop : 1207
      Client Traffic Type : Video Avg Frame Size : 490.649
      Latency (ms) :  Worst latency : 4788.21 avg latency : 412.983 best latency : 7.66921
      Latency (ms) :  97th percentile latency : 511.547
      Latency (ms) :  99th percentile latency : 516.763
      Simulator Start Time : 3, Simulator Duration : 5
      Worst Jitter (milli sec) : 4298.38 Avg Jitter (milli sec) : 23.9961 97th percentile jitter : 28.8205
      99th percentile jitter : 37.4124
      Throughput : 543.639 kbps,   Distance from AP (mts) : 4.65725
      AP Location : (0, 0, 0)                     Station Location : (4, 2, 1.3)
      Average Per : 0
-------------------------------------------------------------------------------------------------------------------------------


-------------------DownLink-------------------
(DS or Edge Load/Total Load) : nan percent and Time Duration : +3000000000.0ns
 Total Resource Available : 0 Used Resource : 0 Unused Resource : 0

--------------------UpLink--------------------
(DS or Edge Load/Total Load) : 100 percent and Time Duration : +3000000000.0ns
 Total Resource Available : 0 Used Resource : 239159 Unused Resource : -239159

--------------------Throughput--------------------
 Aggegate Throughput in AP : 2786.45 kbps
```

Figs. 1a-1c: Sample output from Cisco Wi-Fi 6 simulator

After further research to determine the most relevant independent variables for this study [12][13], it was decided that Modulation Control Scheme (MCS) index, channel width, and guard interval (GI) would be the most appropriate parameters to investigate [14].

The MCS index dictates the number of valid bits in the coding scheme compared to bits used for data confirmation and the modulation that each version uses. Changing the modulation allowed for more data to be transferred, but requires more precision as a result of tighter thresholds[16]. Phase Shift Keying (PSK) allows for different phases to represent different values; using this, we can transmit a single signal wave and change the phase to be read as 1 or 0 depending on whether the phase was 0 or 180 degrees [18], which is called BPSK or Binary Phase Shift Keying. Quadrature Phase Shift Keying improves on BPSK, allowing for 4 different phases which are each assigned a value of 2 bits (00, 01, 10, 11). The more advanced modulation schemes not only change the number of acceptable phases that can be read but also change the amplitude, resulting in Quadrature Amplitude Modulation (QAM) with a number preceding the modulation detailing the number of different encoded phase/amplitude states [18]. As both the coding scheme ratio and modulation complexity increase, the overall throughput should increase as well due to the more valid data being sent at a time, as there are fewer error checks and a larger number of bits that can be sent at once [14].



						OFDM (Prior 11ax)					
MCS Index			Spatial Stream	Modulation	Coding	20MHz		40MHz		80MHz	
HT	VHT	HE				0.8μs GI	0.4μs GI	0.8μs GI	0.4μs GI	0.8μs GI	0.4μs GI
0	0	0	1	BPSK	1/2	6.5	7.2	13.5	15	29.3	32.5
1	1	1	1	QPSK	1/2	13	14.4	27	30	58.5	65
2	2	2	1	QPSK	3/4	19.5	21.7	40.5	45	87.8	97.5
3	3	3	1	16-QAM	1/2	26	28.9	54	60	117	130
4	4	4	1	16-QAM	3/4	39	43.3	81	90	175.5	195
5	5	5	1	64-QAM	2/3	52	57.8	108	120	234	260
6	6	6	1	64-QAM	3/4	58.5	65	121.5	135	263.3	292.5
7	7	7	1	64-QAM	5/6	65	72.2	135	150	292.5	325
	8	8	1	256-QAM	3/4	78	86.7	162	180	351	390
	9	9	1	256-QAM	5/6	N/A	N/A	180	200	390	433.3
		10	1	1024-QAM	3/4						
		11	1	1024-QAM	5/6						

Fig. 2: Modulation Coding Scheme (MCS) parameters vs. Data Rate (Mbps) [9]

The channel width and guard interval describe how much of the frequency spectrum is used for a Wi-Fi signal and how often. For more crowded areas like a shopping mall, 20MHz is often used as it is less likely to interfere with other signals. In contrast, less populated areas can make use of a higher bandwidth up to 80MHz, allowing for higher speed signals to be transmitted at a time. As shown above, an increase in channel width is directly proportional to increased throughput. The guard interval dictates the amount of time between Wi-Fi transmissions needed to prevent interference from reflecting signals. With a lower guard interval the Wi-Fi host can send and receive data more frequently, but it risks higher interference [19].

These three independent variables were deemed to be sufficient in allowing us to verify the results found in the MCS table [14]. Changing the independent variables was accomplished by either directly editing the value in the simulation code, `wifi-ofdma-multi-traffic.cc`, or by using the provided optional command line arguments.

To account for random error, the mean throughput and latency were calculated across clients of the same type for each usable combination of the independent variables. A bash script, as attached to the code listing in the appendix was used to automate the data collection process.

4. Discussion and Conclusion

Discussion

Despite lengthy analysis and reverse-engineering of the Cisco codebase [11], changing the guard interval, channel width, and number of spatial streams did not affect the various simulated clients' performance. Only changing the MCS index via the command line input yielded appreciable results: as expected, as the MCS index increased, the throughput generally increased while latency decreased. These patterns can be observed at a high-level in Figs. 3 and 4.

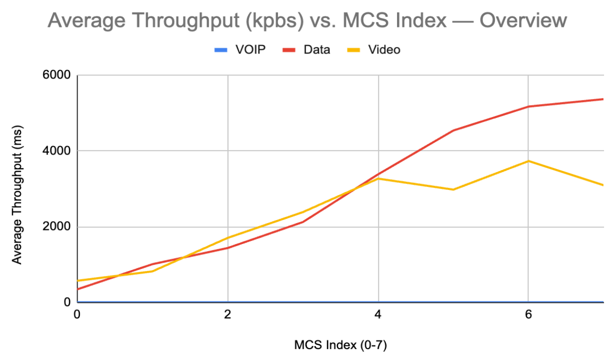


Fig. 3: Average Throughput (kbps) vs. MCS Index — Overview

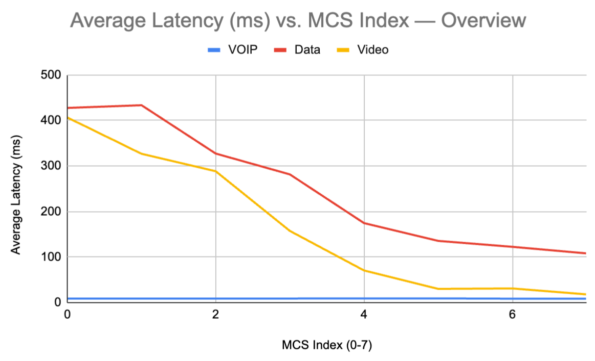


Fig. 4: Average Latency (ms) vs. MCS Index — Overview

To provide sufficient data for this project, only MCS indices from 0-7 were tested. Attempting to use higher MCS values from 8-11, as listed in the sample MCS table, caused segmentation faults during runtime. This is presumed to be the result of these coding schemes requiring the use of the 5GHz radio instead of the default 2.4GHz one, in addition to several undocumented fixes to the base simulation code.

Figs. 5-10 categorize the throughput and latency metrics by client type. In order, we present the average throughput and average latency graphs for the VoIP client, data client, and video client.

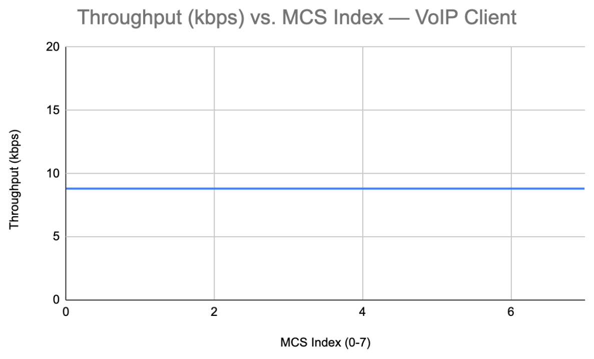


Fig. 5: Throughput (kbps) vs. MCS Index (VoIP Client)

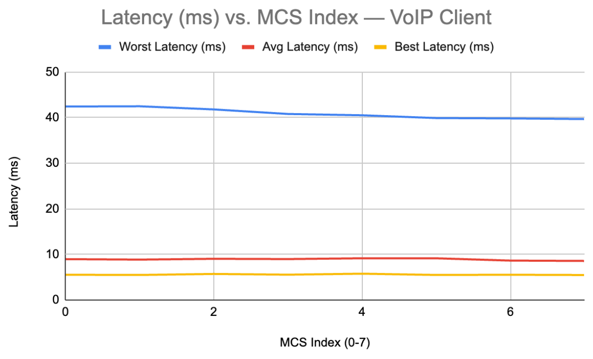


Fig. 6: Latency (ms) vs. MCS Index (VoIP Client)

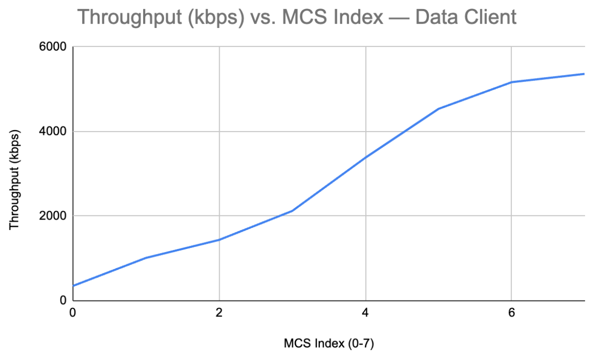


Fig. 7: Throughput (kbps) vs. MCS Index (Data Client)

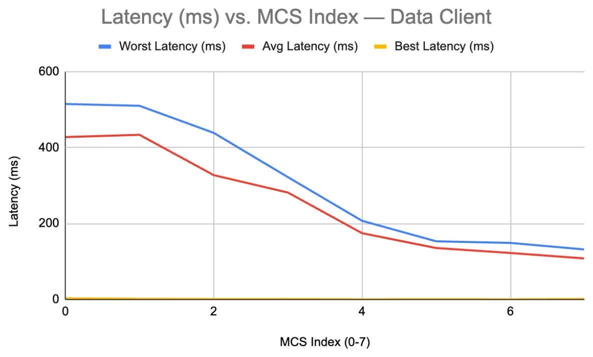


Fig. 8: Latency (ms) vs. MCS Index (Data Client)

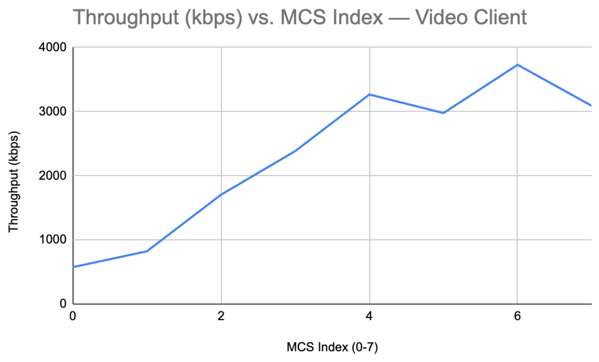


Fig. 9: Throughput (kbps) vs. MCS Index (Video Client)

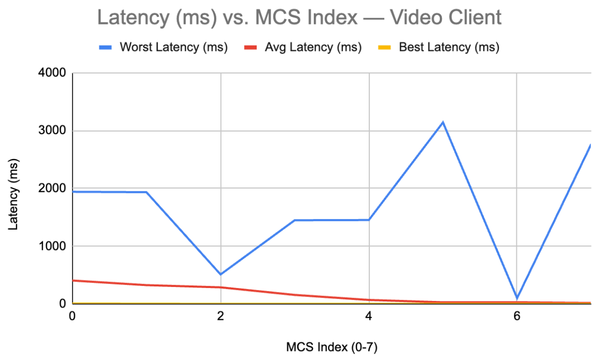


Fig. 10: Latency (ms) vs. MCS Index (Video Client)

While we would expect from the MCS table that throughput and latency should be linearly related to the MCS index, the simulated performance of each client can be explained by how they uniquely interact with the Internet's transport layer [15]. In a VoIP client, throughput and latency are relatively constant, as the protocol's emphasis on reliability and low packet size does not require the highest speeds possible [15]. File transfers and real-time video, however, have elastic throughput and therefore will take advantage of the increased data rate associated with an increased MCS value [15]. The differentiating factor between the data client and the video client is in their latency, as file transfers are not time-sensitive, while video streaming must constrain latency within the range of a few milliseconds [15].

Unlike the MCS index, it was more difficult to find the variables representing the guard interval and channel width within the simulation. To locate these variables, we first explored the showcased models and example commands that were implemented by Cisco from the repository [11], ranging from "HEWiFiPhyHelper", which is used to manipulate physical parameters for the simulation, to "HEWiFiChannelHelper" for editing channel width and other related parameters. The command line inputs did not work and were subsequently changed in our bash script, as we found that by directly specifying their model location, we could still change the values to fit our simulation requirements, or at least display what their model's attributes were.

In practice, attempting to change the guard interval and channel width values did not affect the quantitative results. When modulating the channel width, there was no difference in throughput or latency between any of the clients, despite the expectation that increasing channel width should improve performance. Similar results were found when directly editing the simulation's guard interval variable.

As we had come to find out, this entire repository was based on a multitude of older Wi-Fi versions which is best shown through the ofdma.patch file. Many of the variables that were used in constructing the Wi-Fi simulation were reused from a Wi-Fi 5 simulation which was also carried on from a previous Wi-Fi 4 simulation. Multiple core variables were slightly edited versions of a "yans" (Yet Another Network Simulator) ns-3 Wi-Fi component [16].

While being based on an older Wi-Fi model is not necessarily a bad thing, this specific code base did not allow for MCS indices outside the range of what would be available in Wi-Fi 4, hence the maximum MCS index being only 7 instead of 11. Additionally, the guard interval was used but was permanently fixed to a set size of 0.4 μs.

Despite these shortcomings, we still completed the overall goal of our project as this is an analysis of Wi-Fi 6. Even though some components did not work, we covered our understanding and analysis through external research of related activities. For future work, the most productive change would be using a different existing ns-3 Wi-Fi 6 simulation setup or creating our own using the latest modules provided in ns-3.

Conclusion

In summary, we present a simulation of a Wi-Fi 6 network with VoIP, data transfer, and video streaming clients connected to a single access point. We verified that increasing the MCS index generally has a positive impact on client performance. Due to the limitations of the simulation environment, it was not possible to determine the effect of other variables like channel width and guard interval on performance. For future work, we propose changing the network topology by adding more clients and access points, adjusting the node position, or developing a custom simulation based on the latest ns-3 builtins instead of extraneous code repositories.

5. References

[1] N. Lawrence, “What is WiFi 6?,” Reviews.org. Accessed: Feb. 06, 2025. [Online]. Available: https://www.reviews.org/au/internet/what-is-wifi-6/

[2] E. Tokhirov and R. Aliev, “Analysis of the differences between Wi-Fi 6 and Wi-Fi 5,” *E3S Web of Conf.*, vol. 402, p. 03020, 2023, doi: 10.1051/e3sconf/202340203020.

[3] E. Mozaffariahrar, F. Theoleyre, and M. Menth, “A Survey of Wi-Fi 6: Technologies, Advances, and Challenges,” *Future Internet*, vol. 14, no. 10, Art. no. 10, Oct. 2022, doi: 10.3390/fi14100293.

[4] R. Liu and N. Choi, “A First Look at Wi-Fi 6 in Action: Throughput, Latency, Energy Efficiency, and Security,” *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 7, no. 1, p. 25:1-25:25, Mar. 2023, doi: 10.1145/3579451.

[5] Experimental evaluation of IEEE 802.11ax - low latency and high reliability with Wi-Fi 6? | IEEE conference publication | IEEE xplore, https://ieeexplore.ieee.org/document/10001475/ (accessed Mar. 3, 2025).

[6] Memoona and S. W. Kim, “Performance and simulation analysis of 802.11ax ofdma in contention-driven scenarios,” *PeerJ. Computer science*, https://pmc.ncbi.nlm.nih.gov/articles/PMC11888924/ (accessed Apr. 7, 2025).

[7] Implementation and evaluation of IEEE 802.11ax channel sounding frame exchange in NS-3 | *proceedings of the 2023 workshop on NS-3*, https://dl.acm.org/doi/10.1145/3592149.3592152 (accessed Apr. 7, 2025).

[8] M. D. Hossain et al., "Comparative performance analysis of the IEEE802.11ax and 802.11ac MIMO link for WLANs," *Int. J. AdHoc Netw. Syst.*, vol. 13, no. 4, pp. 1–15, Oct. 2023. [Online]. Available: https://www.ijans.org/ijans-vol-13-no-4-october-2023/

[9] M. Natkaniec and M. Kras, "An optimization of network performance in IEEE 802.11ax dense networks," *Int. J. Electron. Telecommun.*, vol. 69, no. 1, pp. 169–176, 2023. [Online]. Available: https://ijet.pl/index.php/ijet/article/view/10.24425-ijet.2023.144347

[10] L. Louis and U. Surendranathan, “Wi-Fi 6 : performance analysis of IEEE 802.11AX Wi-Fi 6 : performance analysis of IEEE 802.11AX,” 2019. Accessed: Apr. 18, 2025. [Online]. Available: https://louis.uah.edu/cgi/viewcontent.cgi?article=1293&context=uah-theses

[11] Cisco, “Cisco/NS3-802.11ax-simulator: NS3 simulator of 802.11ax,” GitHub, https://github.com/cisco/ns3-802.11ax-simulator/tree/master (accessed Mar. 2, 2025).

[12] Eliud Momanyi Manyinsa, “Implementation of IEEE802.11 (Wi-Fi) in NS-3,” *JCC*, vol. 13, no. 6, Jun. 2016, doi: 10.17265/1548-7709/2016.06.003.

[13] Deronne *et al.*, “ns-3 Wi-Fi 11ax Project: Release final,” May 2019. Available: https://wp.ece.uw.edu/wp-content/uploads/sites/36/2018/11/11ax-final-report.pdf

[14] B. Team, “How to improve MCS index / MCS rates / MAX data rate possible of WIFI Connection,” How to Improve MCS Index / MCS Rates / Max Data Rate Possible Of WiFi Connection, https://www.accessagility.com/blog/how-to-improve-mcs-index (accessed Mar. 2, 2025).

[15] J. F. Kurose and K. W. Ross, Computer networking: a top-down approach, Eighth edition. Hoboken, NJ: Pearson, 2021.

[16] “3: Ns3::yanswifichannel class reference,” ns, https://www.nsnam.org/docs/release/3.18/doxygen/classns3_1_1_yans_wifi_channel.html (accessed Apr. 17, 2025).

[17] Candelatech, https://www.candelatech.com/courses-2023/Session2c_notes.pdf (accessed Apr. 19, 2025).

[18] “Phase shift keying,” ece.unb.ca, https://www.ece.unb.ca/tervo/ece4253/qpsk.shtml (accessed Apr. 18, 2025).

[19] R. Chandra and R. Mahajan, Microsoft, https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/ChannelWidth.pdf (accessed Apr. 19, 2025).

6. CONTRIBUTIONS

- 1. References and literature review
  - 33% Kurtis Lew - Provided multiple references and cited them accordingly, especially a thorough investigation of the Cisco Wi-Fi 6 simulation website
  - 33% Katherine Lee - Provided the most references and communicated details to groupmates for project development
  - 33% Owen Coukell - Provided multiple references and cited them in the report
- 2. Project website
  - 10% Kurtis Lew - Confirmed Website, Assisted in Editing
  - 85% Katherine Lee - Developed Website Shell, Uploaded Text
  - 5% Owen Coukell - Wrote Abstract
- 3. Simulation scenarios, implementation, analysis, and discussion of simulation results
  - 50% Kurtis Lew - Performed various tests, shared output with team and proposed future edits to facilitate project development, Tested additional variables, Graphed output data
  - 20% Katherine Lee - Assisted in finding simulation code, Graphed outputted data
  - 30% Owen Coukell - Assisted in finding simulation code, Tested additional variables, Graphed output data
- 4. Project presentation
  - 33% Katherine Lee - Overview of Related Works
  - 33% Kurtis Lew - Simulation Overview, Results, Analysis
  - 33% Owen Coukell - Introduction, Problem Description, Discussion
- 5. Written interim report
  - 22.5% Kurtis Lew - Introduction, References and Research, Cisco simulator initial reports
  - 17.5% Katherine Lee - Introduction, References and Research
  - 60% Owen Coukell - Abstract, Introduction, Main Section, References and Research, Discussion and Conclusion, Contributions
- 6. Written final report
  - 40% Kurtis Lew - Introduction, References and Research, Cisco simulator final reports, Main section, Discussion and Conclusion
  - 40% Owen Coukell - Abstract, Introduction, Main Section, References and Research, Discussion and Conclusion, Contributions
  - 20% Katherine Lee - Introduction, Related Works, References and Research, Simulator Graphing

7. APPENDIX

Code Listing

The entire base code for this project [8] was made by Cisco and we edited and explored this simulator extensively. The code listings below are points of interest with brief explanations of what we changed and what those changes do, alongside other code segments that didn't work as we'd hoped or had shortcomings that are worth highlighting.

MCS Index Input:

```
CommandLine cmd;

cmd.AddValue ("phyMode", "Wifi Phy mode", phyMode);
cmd.AddValue ("voipInterval", "interval (seconds) between packets", voipInterval);
cmd.AddValue ("verbose", "turn on all WifiNetDevice log components", verbose);
cmd.AddValue ("runNumber", "the index of the run when running from python script",
runNumber);
```

This section present in the main function of the `wifi-ofdma-multi-traffic.cc` file indicates the command line arguments that allowed us to change the MCS index without needing to edit the base file. For our simulation needs, we developed a bash script to run the WiFi 6 simulator with all eight valid MCS indices.

MCS bash script:

```
#!/bin/bash

./waf --run "scratch/wifi-ofdma-multi-traffic --phyMode=HeMcs0
--ns3::WifiPhy::ChannelWidth=20" &> ~/ensc427/output/mcs0_cw20_rev2.txt
./waf --run "scratch/wifi-ofdma-multi-traffic --phyMode=HeMcs1
--ns3::WifiPhy::ChannelWidth=20" &> ~/ensc427/output/mcs1_cw20_rev2.txt
./waf --run "scratch/wifi-ofdma-multi-traffic --phyMode=HeMcs2
--ns3::WifiPhy::ChannelWidth=20" &> ~/ensc427/output/mcs2_cw20_rev2.txt
./waf --run "scratch/wifi-ofdma-multi-traffic --phyMode=HeMcs3
--ns3::WifiPhy::ChannelWidth=20" &> ~/ensc427/output/mcs3_cw20_rev2.txt
./waf --run "scratch/wifi-ofdma-multi-traffic --phyMode=HeMcs4
--ns3::WifiPhy::ChannelWidth=20" &> ~/ensc427/output/mcs4_cw20_rev2.txt
./waf --run "scratch/wifi-ofdma-multi-traffic --phyMode=HeMcs5
--ns3::WifiPhy::ChannelWidth=20" &> ~/ensc427/output/mcs5_cw20_rev2.txt
./waf --run "scratch/wifi-ofdma-multi-traffic --phyMode=HeMcs6
--ns3::WifiPhy::ChannelWidth=20" &> ~/ensc427/output/mcs6_cw20_rev2.txt
./waf --run "scratch/wifi-ofdma-multi-traffic --phyMode=HeMcs7
--ns3::WifiPhy::ChannelWidth=20" &> ~/ensc427/output/mcs7_cw20_rev2.txt
```

This script was also used to change the channel width value in the WiFiPhy module.
As mentioned previously, changing the channel width value to a value other than 20MHz did not affect the quantitative results; we have included the default channel width value for posterity.