Testing Wi-Fi Upgrades for Latency and Throughput

Evaluating OFDMA for Latency Improvements

A Technical Paper prepared for SCTE by

Matt Brooks
Distinguished Software Engineer
CommScope
3871 Lakefield Dr. Suwanee, GA 30024
Matt.Brooks@commscope.com

David Williams
Senior Staff Software Engineer
CommScope
3871 Lakefield Dr. Suwanee, GA 30024
David.Williams@commscope.com

Table of Contents

List of Figures

List of Tables

1. Abstract

Subscriber satisfaction is increasingly about delivering services with the right latency and throughput characteristics. So, how does Orthogonal Frequency-Division Multiple Access (OFDMA) technology deliver this in the home? This paper analyzes empirical testing within a home using typical Wi-Fi clients, focusing on latency and throughput with real-world traffic patterns.

OFDMA in Wi-Fi 6 and Wi-Fi 6E allows for simultaneous transmissions on all spatial streams for each device and subsequently results in lower latency and less contention of airtime. When using both Wi-Fi 6 clients and legacy clients with OFDMA enabled, the latency and jitter improvements are still seen. Wi-Fi 6E access points and clients fully realize OFDMA latency reduction with 6 GHz greenfield spectrum.

The results of testing empirically focus on latency, throughput, and application performance for each Wi-Fi standard and OFDMA setting and showcase use cases of traffic patterns in a house with different client utilizations. By showing latency for a given client while changing only the access point mode and OFDMA settings, a latency reduction is realized in homogeneous client environments as well as mixed client environments. This paper will explain a method for evaluating OFDMA without expensive test equipment or ideal lab setups while providing decision points for when to invest in Wi-Fi 6 or Wi-Fi 6E access points and when to expect an improvement from enabling OFDMA.

2. Introduction

Subscribers are increasingly dissatisfied with their quality of experience (QoE), using programs and applications that have sufficient bandwidth in the home yet are plagued by a problem they do not fully understand: too much latency. Multiple System Operators (MSOs) traditionally sell speed or bandwidth tiers but not latency tiers. Much of the latency conversation is traditionally regarding the wide area network (WAN) access layer to the internet, whether this is fiber, Data Over Cable Service Interface Specifications (DOCSIS), digital subscriber line (DSL), or Satellite. However, the Wi-Fi connection from a gateway/access point (AP) is a common medium by which users access the internet regardless of the WAN link being used.

Using a Wi-Fi 6 or Wi-Fi 6E AP, with OFDMA support, along with devices that also support Wi-Fi 6 or Wi-Fi 6E, allow for a lower latency access to the shared Wi-Fi medium for a more responsive experience during congested or multi-client situations. Improving one layer of the larger network can have a dramatic increase in the QoE of the end-user. All latency values reported in this paper were tested without a contribution from WAN latency [3] referred to in Table 1, which should be considered in addition to values reported in the test results of this paper.

Table 1 – RTT Latency Added from WAN

Last Mile Connection	Latency Contribution
Fiber	10-20 ms
DOCSIS	15-40 ms
DSL	30-65 ms
Satellite	45-500 ms

OFDMA can be used between an AP and client that both support OFDMA. Improvements in latency are possible in mixed legacy client populations by allowing OFDMA capable clients to have some improved latency while less capable clients are still using legacy Wi-Fi standards and continue to tie up access to the whole channel for part of the time. OFDMA is not a feature that is meant to increase throughput and, in many scenarios, can decrease speeds in the current generation of chipsets and software. Decisions on grouping OFDMA capable clients and other scheduler decisions to prioritize speed or latency over one another, as well as serving legacy clients, can change the total throughput seen with OFDMA enabled, negatively impact latency, or cause variation in test results from run to run. In some cases, the data will demonstrate latency is improved, while in other cases it is about the same or worse.

A real house with OFDMA capable clients using a common Wi-Fi client chipset was used to run a few APs through tests to show the relative difference for the same channel but with different OFDMA settings and AP modes used. While changing the wireless mode between Wi-Fi 5, Wi-Fi 6, and Wi-Fi 6E, several variables were tested. This included OFDMA enabled and disabled, different traffic patterns to a client under test, different channel utilizations from other clients, different packet sizes, and different protocols in the downstream and upstream directions while measuring RTT latency on the client under test. Total throughput was also observed, and although Wi-Fi 6 and 6E increased the MCS data rates to support 1024 QAM, as well as allowed better usage of the spectrum with smaller subcarrier width and more efficient use of the spectrum for data subcarriers, throughput was not a focus of the paper.

Results in this paper will show that as you increase the channel utilization or the throughput needs of the client under test, both conditions are more conducive to showing a decrease in latency if OFDMA is enabled. The reduction in average round trip time (RTT) latency is sometimes seen but more often a dramatic decrease in the maximum RTT latency measured is observed with OFDMA enabled. This is also accompanied with a lower median deviation of RTT or variance of the latency known as jitter. This means that even in cases where the average RTT is about the same, the maximum and median deviation RTT are often meaningfully lower providing a smoother QoE for the end-user with applications less likely to encounter latency spikes that cause issues with the QoE.

3. Latency Overview

Much effort is put into emphasizing network throughput achieved or bandwidth available, but latency is also one of the most important and often overlooked metrics. A consistent and good QoE reduces churn for MSOs, and latency is a main contributor to QoE. A user may be able to achieve the same throughput, that is data over time, on different networks, but one network could have less latency and be more responsive and pleasant to use.

When discussing latency, it’s important to define terminology. This paper will discuss round trip time (RTT) vs. one-way delay (OWD), as well as the difference between idle pings and pings under load, and lastly a consideration for RTT of each transmission control protocol (TCP) stream or session vs. RTT of a ping to and from the source and destination.

Latency means different things to different people even in the networking industry. End-users outside the networking industry are that much further removed from knowing what is good or bad and may blame the wrong metric for a poor user experience. Latency can be defined simply as the amount of time for information or a packet to arrive from a source to a destination and is comprised of components such as: propagation delay (time required for the signal to travel over the medium), transmission delay (time required to push the bits into the link), processing delay (time required to process packet headers), and queuing delay (amount of time a packet is waiting in queue to be processed and gain access to the physical medium) [1]. This can be defined over a subset of network segments or a full network delay of a packet going to the destination over multiple hops locally or through the internet. This delay is also different under a working load vs. while idle. Latency observed in an idle network or idle Wi-Fi channel does not correlate with the user experience on the same network under a different load. On the wireless medium, latency is an especially fluid number changing constantly based on traffic passed, the number of clients accessing the shared medium, un-coordinated channel collisions, distance, and retransmissions.

Latency can be defined by the time it takes for a packet to be sent from a source and received at a destination; this is referred to as OWD. OWD measurements are usually accomplished with user datagram protocol (UDP) and require clock synchronization between source and destination to measure the elapsed time without having feedback to the source's clock. A timestamp is added in the data of the outgoing packet which is measured against the destination's clock when it is received. For this method of OWD measurements to be accurate, the same synchronized time must be used on source and destination devices. One such tool that uses this approach is an application called NUTTCP when used with UDP.

Latency can also be defined by timing how long it takes for a packet to go from source to destination and back to source with a response. This method is useful as well because most packets in networks are going to a destination to solicit a response back to where it came from. This may not be latency in its purest sense, but it is a total latency that most users would experience with a real application. This full elapsed time is measured and referred to as round trip time (RTT). Internet control message protocol (ICMP) pings and TCP RTT are common ways of measuring RTT. Pings over long multi-hop connections are not necessarily a true reflection of the latency in a loaded network, because network equipment could choose to prioritize or de-prioritize handling ICMP pings or cached information in routers could speed up subsequent pings. However, on a local network with a single hop, pings are still an essential tool to measure RTT delay. Iperf3, a networking throughput testing tool, can also report RTT per TCP flow while data is being sent if sourcing from a Linux Ethernet port.

3.1. Latency Values and Test Techniques

Considering only the instantaneous or just the average latency measured is not enough to characterize QoE of the end-user in a real wireless network. In addition to the average values of RTT or OWD, consideration of the minimum, maximum, and jitter, or a measure of the variability of the latency reading during a measurement period, is necessary. Sometimes, the average or minimum latencies won't change much in test scenarios, but the maximum latency observed can be far higher than the average and responsible for problems in user experience [1]. Many applications can handle a higher consistent latency, but if a packet suddenly has a much higher delay than average, this can cause extremely noticeable problems depending on the application being used. If an application can buffer without the user noticing, such as video playback, the end-user may not notice high jitter or higher latencies as easily.

Consider the difference in effect of high latency packets while watching a movie that buffers a significant amount, a voice call or video call that buffers very little, a virtual reality headset tracking a person's movements, or online game play that experiences a sudden skip in graphics displayed or a delayed reception of an action of a player. General expectations for good and bad latency values per application type are found in Table 2 below [4,5,6]. This includes full network round trip time latency including WAN, while the values in this paper are the latencies inside the house with Wi-Fi and do not include a WAN component.

Table 2 - Common Application Latency Expectations for RTT Latency

Application	Excellent	Good	Fair	Bad	Recommended
Gaming (FPS)	<20 ms	20-50 ms	50-150 ms	150+ ms	x
Gaming (MMO)	<20 ms	20-100 ms	100-250 ms	250+ ms	x
Gaming (RTS)	<20 ms	20-100 ms	100-200 ms	200+ ms	x
Cloud Gaming *	<20 ms	20-30 ms	30-40 ms	40+ ms	x
Voice Call	x	x	x	x	<100 ms
Video Call	x	x	x	x	<100 ms
Video Stream	x	x	x	x	<100 ms

* Latency requirements for cloud gaming are more stringent due to control input

A loaded network, especially in wireless networking, will have increased delays when traffic is being sent and while serving multiple wireless clients on the shared medium. Enhanced distributed channel access (EDCA) provides the mechanism for how backoffs are observed between different APs and clients trying to access the same channel. The more clients attempting to use the channel the greater the chance for one of them to need to backoff and wait even longer before winning an opportunity to transmit on the channel.

A network with traffic during tests or pings is referred to as experiencing load or being loaded. For example, a Wi-Fi 5 network with just 4 clients and very low utilization on the channel of 10% can still achieve 2 to 5 ms RTT pings with a very low 0.5 Mbps iperf3 data flow in the downstream direction. However, the same very low 0.5 Mbps TCP data flow to a Wi-Fi 5 client while the channel is 90% loaded, causes the same flow to incur an average of 19 to 24 ms RTT pings, up to a max of 187 to 198 ms possibly. A loaded network's delay is a more accurate view of the worst-case latency experienced by an end-user and is a preferred situation to characterize latency improvements. Loaded networks have more delay in queues/buffers, from backoff timers for fair access of other clients, and from retransmissions. Without load, it is more difficult to make discernable differences in outcome or QoE with different technology or settings when later used in a real environment.

In Wi-Fi channels, for this paper, the load refers to channel utilization which does not refer to Mbps. Channel utilization could be very high, even for a very low Mbps being transferred, if a slower, more robust MCS rate is being used for a client that is far away. Referring to load as airtime utilization abstracts away problems in test setups that specify loads as Mbps which are not the same anywhere else. Scheduler algorithms in AP chipsets are proprietary and have thresholds for considering usage of OFDMA to begin with, and idle or low usage clients are often not allocated any resource units (RUs). In this paper, many channel utilizations and throughput levels to a client under test were characterized.

The above methods for determining latency are simplifying what can be complicated. The absolute RTT or absolute OWD may change with a different number of TCP flows or different TCP window sizes, better time synchronized clocks, and with special treatment of ICMP pings by each operating system. In addition, if separate Ethernet links were used to source to each client, further differences may be observed. However, the relative differences between AP Wi-Fi modes and OFDMA settings, while keeping test methods and conditions the same accompanied with automation of test execution, allows for compelling comparisons.

4. OFDMA Overview

The primary feature being tested empirically in this paper is OFDMA. This technology first available for 802.11 networks in the Wi-Fi 6 standard allows for simultaneous transmissions to and from clients in the same channels on different subcarriers. This contrasts with previous generations of Wi-Fi standards, which are for the most part round robin transmissions. Multi-user multiple input multiple output (MU-MIMO) in Wi-Fi 5 was able to achieve simultaneous transmissions, but it was not able to predictably realize gain in many real environments and often causes too much self-degradation and overhead to be beneficial especially with more than 3 clients. It did however improve latency in many cases even with increased retransmissions or lower throughput as client counts increased. To isolate contributions of OFDMA to latency, MU-MIMO was disabled for very high throughput (VHT) and high-efficiency (HE) modes in downlink and uplink directions.

OFDMA is often presented in settings of the web user interface of an AP with control separated to enable downlink OFDMA, that is AP to client, as well as uplink OFDMA, that is client to AP. For this paper both downlink and uplink OFDMA will be enabled or disabled together. The AP scheduler, proprietary in each chipset's firmware, is responsible for determining what technology should be used or which groups to form for simultaneous transmissions. For each transmit opportunity (TXOP), the AP will choose if the transmission will be single user (SU) legacy orthogonal frequency-division multiplexing (OFDM) traffic, synchronized multi-user OFDMA uplink traffic or synchronized multi-user OFDMA downlink traffic. The OFDMA modes are referred to as high-efficiency multiple user (HE-MU). This is because of the synchronized aspect of OFDMA transmissions, which drives efficiency, instead of legacy carrier sense multiple access with collision avoidance (CSMA/CA). Each OFDMA capable device is assigned a subset of subcarriers, or tones, by the AP for that device to simultaneously transmit on at the same time as others do on different subcarriers in the same channel.

With OFDM in Wi-Fi standards prior to Wi-Fi 6, the subcarrier spacing is 312.5 KHz, however, with Wi-Fi 6 and OFDMA the subcarrier spacing is 78.125 KHz. The subcarriers are 4 times closer together in OFDMA, and the symbol time is increased to be 4 times longer from 3.2 microseconds to 12.8 microseconds. The reduction in size of each subcarrier also results in efficiency gains in the channel itself because of less spectrum being used for pilot subcarriers and null guard carriers. The trigger frame from the AP to the clients specifies which RUs are allocated for simultaneous uplink transmission. In Wi-Fi 6 on 2.4 GHz and 5 GHz bands, the data exchanges can occur with OFDMA, however, many control and management frames must still use legacy OFDM to hold off and notify legacy clients about the channel being used [2]. The exception is for HE specific control frames such as buffer status report (BSR), clear-to-send (CTS), and block acknowledgements (ACK) which can occur simultaneously on RUs that are assigned.

4.1. OFDMA: Resource Units

Resource units (RUs) are groups of OFDM subcarriers, referred to as tones, and are predefined in these allocations: 26, 52, 106, 242, 484, 996, or 2x996 tones. The location of the RU within the channel is further defined by an RU index contained in a trigger frame that lets the client know what part of the spectrum is set aside for it in the single TXOP. Most current generation APs use a 242 tone RU as the smallest RU allocation in the 5 GHz and 6 GHz bands when 80 MHz channels are in use. A total of four 242 tone RU assignments can exist in a single group if the channel is 80 MHz. The 2.4 GHz band was not considered in this paper, but naturally uses smaller than 242 tone RUs to achieve multi-client OFDMA transmissions in 20 MHz and 40 MHz channels.

The AP can create additional groups to simultaneously transmit or receive from different sets of clients using the same RUs but in different groups, and therefore at different times. Multiple groups using OFDMA within each group are still advantageous to a certain point, but it does reduce the benefits of OFDMA and a study on multiple groups and its effect on latency reduction is left for further study. There would be a certain point when clients have unequal bandwidth needs in a specific TXOP that warrant scheduling a single group with lower RU designations than 242 tones. However, currently 242 tone RUs are preferred most often in simultaneous 4 client scenarios as the AP doesn't have to allocate extra null guard carriers or pilots in the spectrum as it does with smaller 106 tone RU assignments. When 106 tone RUs are assigned to be used in an 80 MHz channel, there are 8 to 13 total clients supported in the group if the AP allocates 26 tone RUs to 5 of the clients. Some APs are not allocating smaller RUs with larger RUs in the same group and would leave gaps in the spectrum unused. However, using RU assignments of less than 242 tones is more advantageous for latency reduction with many clients at the same time. In Figure 1 below, the difference between single-user and multi-user OFDMA is shown with respect to time; one or more clients will be allocated tones across an 80 MHz channel. This example shows 996, 484, and 242 tones being used over time.

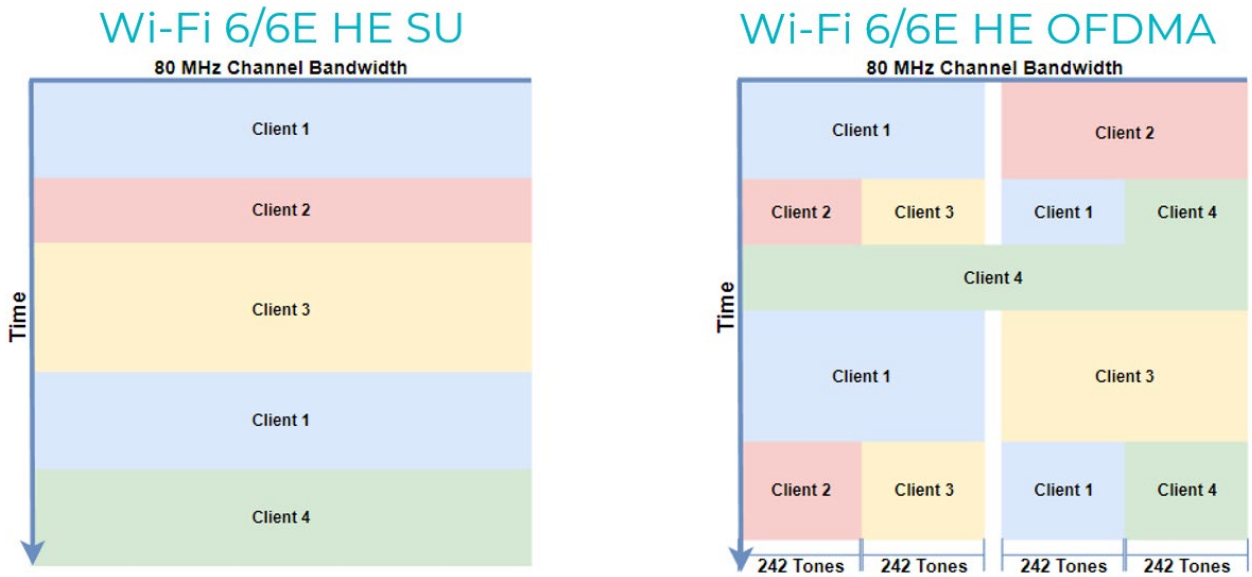

Figure 1 – Example OFDMA Tone Allocation with HE SU vs. HE OFDMA

The AP's scheduler is constantly managing if it prefers two clients at 484 tone RUs each in two separate groups to alternate between, or if it prefers to have four clients in a single group of 242 tone RUs each. This was observed to constantly change and is unknown exactly what proprietary reasons, other than the amount of traffic needing to be sent, that the AP is using to make decisions to use 2 groups of 484 tone RUs each or a single group of 242 tone RUs each. Receive power differences per client is one such reason for an AP to prefer to group certain clients together. However, when seeing spontaneous and rapid changes in RU assignments while clients are physically static and traffic is at a set limit, the rapid RU assignment changes and group changes are something that should be improved upon and will as AP schedulers are matured.

4.2. OFDMA: Process

For downlink OFDMA, non-Wi-Fi 6 clients are aware of the time they must be silent because after the AP has won a TXOP, it will send a multi-user request-to-send (MU-RTS) frame to clear the channel for the length of time of the full OFDMA exchange. The physical header and mac layer will contain the RU assignments, but they can also be specified in other trigger frames, or buffer status report frames. Other Wi-Fi 6 clients receive this frame and respond with their own CTS on their designated RUs at the same time. Next, the AP will send simultaneous data or multi-user downlink physical layer convergence procedure (PLCP) protocol data units (MU DL-PPDU) on each client's RUs. Clients will auto block ACK or wait for a block ACK request (BAR) from the AP and send upstream a block ACK at that time [2].

For uplink OFDMA, the AP still must win a TXOP, and only then can it schedule simultaneous uplink transmissions with uplink OFDMA on the clients that support it. A buffer status report poll (BSRP) is sent from the AP to the clients and solicits a BSR from each client. The BSRP contains the RU designations to be used for each client to respond with its own BSR which contains information about how much and what quality of service (QoS) of data needs to be sent upstream. This BSR information can also be unsolicited and indicated to the AP using a QoS control field in a data frame. A MU-RTS trigger frame may be sent from the AP and assigns RUs to each station as well as serves to notify the legacy clients of the upcoming transmission. The AP then waits for a CTS to return on each clients' assigned RUs. The MU-RTS is optional and can be skipped; the AP can go straight to a basic trigger frame which contains information about which RUs each client can use, the power each should try to use, as well as spatial streams and MCS rates to send their data upstream simultaneously. Next, the uplink PLCP protocol data unit (UL-PPDU) from each station is received for the same amount of time and each client will pad data if there is empty time. Finally, a multi-STA block ACK is broadcast so all clients can discover which frames need to be resent and the AP may optionally choose to send individual block ACKs to each client [2].

The goal of any network is to maintain high enough throughput and low, consistent latency, and low packet loss to provide a smooth, responsive, and predictable user experience. OFDMA technology in Wi-Fi 6/6E provides a mechanism to realize decreased maximum and average latencies in the Wi-Fi layer, as well as decreasing the variance or range of latencies experienced in more heavily utilized Wi-Fi channels.

5. Test House Setup

The Wi-Fi test house used for this testing is over 4500 square feet and consists of 3 stories including the finished basement and typical build materials such as wood floors, carpeted floors, sheetrock walls, and furniture throughout; it also has clean channels available with 0% utilized airtime because of the distance to neighbors. Microsoft Windows test clients were used to represent a more typical use case and good control of clients. The AP location was in a front corner of the house.

In contrast to normal lab testing techniques that seek to setup ideal conditions including precise location of clients, angles that present equal power to and from each client, and intricate test instrumentation, this paper was meant to determine if improvements can be realized in normal and non-curated test conditions with easy-to-use test applications. The locations of clients for this test were also chosen to represent a normal use case of clients in simultaneous use in the following locations: a client on the 2nd floor above the AP, a client in the basement below the AP, and a client in the same room as the AP.

A fourth client, the client under test (CUT), was located 26 ft away on the same floor and was the farthest client in this test on purpose as end-users are not usually right next to the AP. This client was used to ping and monitor more closely to investigate when the client was able to see better latency or not. The client was also used to play the game on for the test case involving playing an online game. This client under test would represent an end user's experience while other clients are using the same Wi-Fi network at various loads. The AP was not rotated to find ideal angles since comparison tests were planned, and therefore relative differences only were being considered.

6. Test Tools

Many test tools and scripts were evaluated including subscription-based tools we routinely use in our lab environments that are not freely available. However, there is value in being able to show improvements with tools that are freely available, such as iperf3. Another free tool NUTTCP was considered, but we had problems with ensuring time synchronized clocks with a LAN NTP server in the Wi-Fi test house.

This negated the advantage to using NUTTCP with UDP to measure OWD as this tool can do with well synchronized clocks. Also, because NUTTCP did not show the RTT of each TCP stream and instead only found RTT before the test started, it would have necessitated more complexity by using a second tool to get TCP RTT results. IxChariot was also considered but it proved harder to allow the type of automation required to test and then send back certain values into the next test automatically.

When using a source computer running Ubuntu Linux combined with iperf3, we were able to get RTT per TCP flow in the downstream direction, AP to Client, and this proved valuable enough to combine with ping data to select iperf3 as a tool. In the upstream direction, since the clients in use for this paper were Microsoft Windows to show applicability to real world use cases, we were missing the ability to get per TCP stream RTT. However, we still had the ability to run data in the upstream direction and use pings to evaluate the channel.

Another appealing reason to select iperf3 was the easy-to-use javascript object notation (JSON) output for aiding use in automation. The large number of test iterations planned called for creating some automation to organize and execute tests and remove any possibility for user error in setup of test parameters. Python3 was used to automatically setup certain iperf3 settings and rates, get data, and calculate different rates based on the data from a prior baseline run. Python3 also allowed for easy parsing of JSON output and presenting data in an easy to consume format, even the exact format needed to organize data in Excel. Using a Linux Ubuntu 20.04 operating system source Ethernet on LAN allowed for more precise timing control of pings used at a rate of 10 per second and was started just after traffic began and terminated just before traffic ended. The following network config parameters were changed on the linux machine to allow a larger TCP window size: *net.core.optmem_max = 524287; net.ipv4.udp_rmem_min = 8192; net.ipv4.udp_wmem_min = 8192; net.core.rmem_max = 16777216; net.core.wmem_max = 16777216; net.core.rmem_default = 2097152; net.core.wmem_default = 2097152; net.ipv4.tcp_rmem = 4096 87380 16777216; net.ipv4.tcp_wmem = 4096 87380 16777216; net.ipv4.tcp_mem = 1638400 1638400 1638400*. Using python3 allowed precise control of iperf3 sessions to all clients while simultaneously saving the TCP RTT data and using controlled pings to gather RTT information and throughput information for each controlled test cases.

The version of iperf used was 3.9. An example of the iperf3 commands that the custom written python3 scripts would launch with ‘Popen’ simultaneously and receive JSON output from is seen below. The rate-limited bitrate is per flow, and in this example, it was 4 flows per client at the bitrate listed in -b. The first 4 seconds of each run were always discarded with the -O omit flag. -J command was used to specify to receive results in JSON format. -R --get-server-output was used for an upstream direction test to specify receiving data at the Ethernet source from the clients.

- */usr/bin/iperf3 -c 192.168.0.238 -i4 -P4 -w2M -M1460 -b12.5M -t60 -O4 -J -T cut*
- */usr/bin/iperf3 -c 192.168.0.2 -i4 -P4 -w2M -M1460 -b62.13M -t60 -O4 -J -T client2*
- */usr/bin/iperf3 -c 192.168.0.3 -i4 -P4 -w2M -M1460 -b62.205M -t60 -O4 -J -T client3*
- */usr/bin/iperf3 -c 192.168.0.4 -i4 -P4 -w2M -M1460 -b62.3625M -t60 -O4 -J -T client4*

The JSON output received back per client was converted to a simple python3 dictionary object for parsing. The data for each stream to a client was then processed to take the average RTT of the streams, the maximum of the max RTT, and the minimum of the min RTT to represent that client's data in the test. An example of the data contained, per stream is seen in Figure 2.

```
"sender":   {
    "socket":   9,
    "start":    0,
    "end":  59.999983,
    "seconds":  59.999983,
    "bytes":    467795968,
    "bits_per_second":  62372813.405630462,
    "retransmits":  128,
    "max_snd_cwnd": 770880,
    "max_rtt":  32568,
    "min_rtt":  16234,
    "mean_rtt": 21106,
    "sender":   true
```

Figure 2 – Example iperf3 JSON to python3 Dictionary for a Single Flow on One Client

An example ping command, launched one second after the simultaneous iperf3 flows started is: *ping 192.168.0.238 -i.1 -c570 -w57 -D*. The pings were sent at a rate of 10 times per second and were set to end at 57 seconds with -w flag regardless of being able to complete the 570 pings specified by -c flag. This was necessary to make sure any delays or loss did not result in pings being measured after iperf3 flows had ended. The total time of traffic being sent from iperf3 to clients was 64 seconds, with the first 4 seconds being discarded.

Below in Figure 3 is an example at the end of the ping data received during a test; this allowed for parsing and graphing per ping sample.

```
[1656698977.360137] 64 bytes from 192.168.0.238: icmp_seq=560 ttl=128 time=45.0 ms
[1656698977.440025] 64 bytes from 192.168.0.238: icmp_seq=561 ttl=128 time=24.8 ms
[1656698977.542482] 64 bytes from 192.168.0.238: icmp_seq=562 ttl=128 time=26.3 ms
[1656698977.659009] 64 bytes from 192.168.0.238: icmp_seq=563 ttl=128 time=42.3 ms
[1656698977.750418] 64 bytes from 192.168.0.238: icmp_seq=564 ttl=128 time=33.2 ms
[1656698977.862114] 64 bytes from 192.168.0.238: icmp_seq=565 ttl=128 time=44.6 ms
[1656698977.951455] 64 bytes from 192.168.0.238: icmp_seq=566 ttl=128 time=33.1 ms
[1656698978.036393] 64 bytes from 192.168.0.238: icmp_seq=567 ttl=128 time=37.8 ms

--- 192.168.0.238 ping statistics ---
567 packets transmitted, 567 received, 0% packet loss, time 56941ms
rtt min/avg/max/mdev = 6.434/28.538/248.917/20.525 ms, pipe 3
```

Figure 3 – Example Ping Data Received During a Test

A particular online game was chosen as well, Counter Strike Global Offensive (CS:GO), because it was hosted easily on the Ethernet LAN connected Linux server that also sourced iperf3 traffic, and it was able to be played without interaction with the WAN internet at all. This game also has a developer mode that overlays latency and variance on the screen during play. This game is also not graphic intensive and was used only for the networking measurement aspect to show any relevant differences experienced in a real application under different channel load conditions and AP settings.

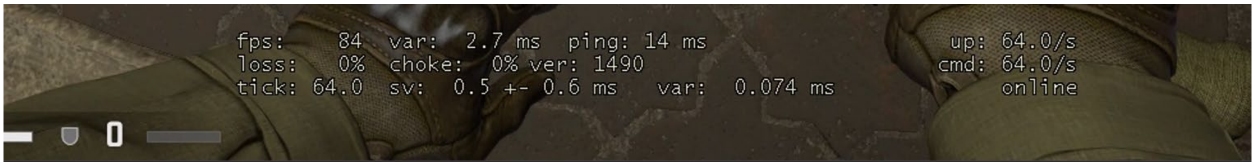

Figure 4 – Counter Strike Global Offensive Screen Overlay with RTT Latency

Figure 4 above shows an example of the overlay used to get the game's measurement on the RTT of data which changes during a given 60 second run. Its value was recorded every 3 seconds for analysis during traffic for the given game play test scenarios. This information along with iperf3 RTT data and ping data was used to characterize latency in this real application while the channel was set to various utilization levels and the AP under test was used in different modes.

7. Test Methodology

7.1. Test Constants

This paper sought to identify which scenarios can show lower latency, lower max latencies, or a more consistent delay with much lower variation in latency. This setup was specifically not in a lab or in a heavily curated test with equal power clients and equal traffic to each client. 2.4 GHz was not considered for this paper. For 5 GHz testing, channel 100 was used as it was a completely clear channel in the test house which allowed for repeatability. For Wi-Fi 6E tests channel 37 was chosen. A channel bandwidth of 80 MHz was used because this paper was about evaluating AP modes, OFDMA, and latency, not throughput. Increasing the bandwidth of the channel to 160 MHz would inflate the total throughput required on each test case to max out the channel and leads to less deterministic or repeatable results. However, some chipsets and firmware today can aggregate even more clients on 242 tone RUs in a single group with 160 MHz. This will be left to one of many additional tests for a later date.

Another test constant decided for this paper was to completely disable MU-MIMO including VHT MU-MIMO as well as HE MU-MIMO. MU-MIMO contributions to latency or throughput was not the focus of this paper and would have only served to confuse the contributions of OFDMA in various test scenarios including when baselining the ability of a channel with three clients, vs. adding in one more client under test in the same channel. The additional client under test added to a scenario would change the interference experienced and grouping of the other clients negating the baseline if MU-MIMO was enabled beforehand. The various non-repeatable conditions and baselining issues in addition to the overhead of null data packet announcement frames and beamforming report poll frames made disabling MU-MIMO for this paper an easy decision.

Default QoS was used by not setting QoS at all, to not intermix this layer of prioritization in with the evaluation of the OFDMA feature in these test conditions and iterations already defined. Other constants included the fixed physical positioning of the AP and clients between representative loads and setting changes. A certain AP was used to ensure coverage of Wi-Fi 6E data, however, the maturity of firmware on the Wi-Fi 6E AP is still lagging APs with just 5 GHz Wi-Fi 6. Two other APs for a good sampling of today's AP abilities were used to test in 5 GHz.

7.2. AP Settings for Comparison

The AP under test was controlled and tested against the following modes, settings, and client modes:

- Wi-Fi 5
- Wi-Fi 6 with OFDMA DL & UL disabled
- Wi-Fi 6 with OFDMA DL & UL enabled
- Wi-Fi 6 with OFDMA DL & UL disabled and two clients set to Wi-Fi 5 (802.11AC) only
- Wi-Fi 6 with OFDMA DL & UL enabled and two clients set to Wi-Fi 5 (802.11AC) only
- Wi-Fi 6E with OFDMA DL & UL disabled (if available)
- Wi-Fi 6E with OFDMA DL & UL enabled (if available)

At the beginning of a test with the AP set to a particular mode and OFDMA setting, a set of baseline tests were executed. The baseline tests determined what the max average rate was for each of the three control clients running traffic together. This was baselined separately for each AP mode and OFDMA setting and directly preceded each set of test iterations listed in the client conditions for comparison section.

The baselining served a couple purposes. Each client was in a different location which represented a more typical use case and therefore each client's capability to achieve a certain Mbps throughput was different. For example, the same throughput achieved in Wi-Fi 6 was more than what was achievable in Wi-Fi 5; it would not be a fair comparison of channel utilization to set the Wi-Fi 6 speeds at the same rate limits used in the Wi-Fi 5 testing, because the channel would be at a different airtime utilization in the comparison. A better approach was to determine what speeds the clients could achieve during each set of AP settings, then discount the rates of those clients to achieve an approximate percentage of channel utilization; this would be a different bitrate for each client and each AP mode, packet size, OFDMA settings, traffic direction, and client mode settings. Another reason for doing this is certain settings in use would cause different schedulers to be employed and different technology or decisions to be used. For example, in UL OFDMA the AP is instructing the clients which MCS to use amongst other parameters, however, with OFDMA disabled, the clients are choosing their own rates.

For these and other reasons it was decided to not rate limit other clients based on Mbps blindly for all test cases, but instead to rate limit based on percentages of max achieved average rates when the other three clients were used simultaneously without rate limits. After determining the throughput on average achieved by each of the 3 clients with no rate limits, this allowed a known discounted rate to approximate channel utilization to a given percentage for subsequent tests. This method should also allow extrapolation of findings to similar test scenarios with clients that are far away and not able to do high rates, but they could of course use up certain percentages of the channel's airtime just the same.

7.3. Client Conditions for Comparison

During the test iterations for all the AP settings defined above, different parameters for the other clients and the CUT were then tested over the following test parameter iterations:

- Downlink and uplink traffic direction
- TCP and UDP
- Client under test rate limits of 500 Mbps, 50 Mbps, 5 Mbps, and 1 Mbps while also playing a low bandwidth LAN Ethernet hosted game.
- Other three clients rate limited to 10%, 50%, and 90% of their capability determined in prior baseline tests with just the three clients running traffic without rate limits
- 536 byte and 1460 byte size payloads (Used for both TCP and UDP tests)

The different rate limits chosen were to represent high, medium, low, and very low rates for the CUT. The rate of 500 Mbps was specifically chosen to be below the full channel capacity for most test conditions, but also higher than what could be achieved in a 242 tone RU assignment for a 2x2:2 client at the client locations chosen. This was to expose some areas where OFDMA can reduce throughput as schedulers struggle to figure out how to divvy up tones and groups for multiclient scenarios with high data rates for some clients but not all clients. The scheduler must decide how to handle a highly utilized client in the face of lower utilized clients. There is not a right or wrong answer here, and we believe it is a source of some unknown expectations and ongoing improvement.

The highly utilized client could be transmitted to using the entire channel and then rotated between the other OFDMA transmissions of the 3 other clients using their RU designations for short periods of time. However, the highly utilized client could also be put in a smaller than needed RU designation for the sake of preserving low latency access to others and put in the same group as the lower RU designated clients; this would cause throughput degradation in the name of reducing total latency for all clients. Finally, it could decide to only briefly use a smaller RU combined in the same group with other lower utilized clients for part of the time, and then switch back to full usage of the channel causing higher latency on the other clients.

The channel utilization percentages chosen included 10% for the other clients to represent low utilization either in the house or to an extent, overlapping neighboring networks on the same primary or secondary channels of a similar percentage. Utilizations of 50% and 90% would represent busy and very busy channels which will uncover at what points of channel utilization OFDMA starts to help with latency reduction. The same testing re-executed while creating controlled amounts of out-of-network traffic on the same primary and secondary channels are left to future testing. This would be to characterize the difference in results with uncorrelated utilization and EDCA backoffs being observed between different APs in other houses or even multi-AP mesh systems sharing channels within a house.

The different packet sizes represent a range of payloads, small and large, to determine if benefits are only seen based on the size of packets sent in the channel. The same payload was chosen for UDP payload and TCP max segment size (MSS) tests for parity's sake even though the UDP data could have been smaller than the lowest TCP MSS and 12 bytes larger than the largest TCP MSS. The packet size for a test scenario is used to all 4 clients for the test case, not just the CUT. iperf3 TCP flows were set with 4 flows except for the CUT at 1 Mbps and 5 Mbps where multi-flow was not beneficial; in this case just 1 flow was used. iperf3 TCP window size was set to 2 MB in all cases. UDP was also tested to isolate much of the other direction's traffic except for MAC layer 2 acknowledgments (control frames) to see if behavior is much different without the constant interruptions to send and receive TCP acknowledgments as data frames in the opposite direction. Lab test scenarios often start with UDP for more exact control, but it is not indicative of real-life application flows this paper sought to primarily evaluate.

8. Test Results

Different AP chipsets were used to do this testing and to investigate what amount of channel utilization or amount of client under test utilization can show improved latency with the OFDMA feature in current generation firmware. The test environment was not overly curated to create ideal lab test conditions in a chamber or test cases that included every client receiving the same low to medium throughput with UDP and equal power as many lab test cases and vendors would and should execute. This paper evaluated a non-lab test scenario in an actual house of four clients of controlled and different throughput with TCP and UDP while at varying AP receive power, AP transmit power, and distance. This simulated a real situation but with repeatable, controlled utilization of clients to facilitate finding relative differences between OFDMA settings and AP modes in a real environment.

The relative comparisons and charts were made on the same chipset for direct comparison and consideration of OFDMA's contributions to latency. There were extreme differences in chipset abilities, algorithms, and scheduler decisions, but comparisons were made on the same AP. 260 to 364 tests created approximately 5,000 to 6,000 data points per AP and protocol. This was executed on three APs with both protocols creating total data points of well over 30,000 between the modes and different APs tested. Each test also had additional raw data saved that represented the per sample data making up the averages, min, and max; some of the per sample data was also analyzed and graphed. The data was examined to find generalizations to illustrate improvements that could be expected in certain scenarios.

There is much data created from this type of exhaustive testing, as well as what would be created from the myriad of proposed follow-up testing scenarios. Many of the scenarios tested were not expected to show a latency RTT improvement. Executing each scenario, however, was the only way to find at what amount of channel usage and traffic level the client under test would start to realize a latency improvement in today's chipsets and firmware. Improvements and more testing will be continuously sought, while drawing more conclusions to the data already collected.

The most compelling RTT latency reduction improvements included the scenarios with high channel utilization, including maxed out, non-rate limited iperf3 flows for each client as well as many of the 90% channel utilization test cases. Also, the higher usage of the client under test provided the highest improvement in latency RTT for that client. The other generalized improvement was when using a larger MSS of 1460 bytes compared to smaller MSS of 536 bytes. Many test results did not show favorable results and in many scenarios, this was expected. For example, in a low utilized channel the airtime is not causing contention and clients already get access to the channel without the need of OFDMA. Test results selected to be shown in this paper are just a very small subset chosen to represent a mix of UDP and TCP, large and small packets, downstream and upstream, different rates to the CUT, and different channel utilizations.

For each chart consider the following way to look at the chart. The categories on the X axis are used to group test results for a given AP mode and client mode scenario including OFDMA enabled or not. Each test result for that given test mode is then listed front to back, starting with latency values as indicated and labeled per parameter as a category to the right, and ends with total throughput for reference in the back of the chart. The Y axis is being use for both Mbps for the last category per group and serving as a millisecond reading for all the other categories with latency.

8.1. Non-Rate Limited Test Scenarios

The tests with no iperf3 rate limits to all four clients were sourced from or destined to a single Ethernet 1 Gbps port on the LAN of the AP. Non-rate limited scenarios are perhaps where many test efforts start and is worthwhile to characterize, but it is not a likely scenario to occur in a house. However, since this test case is craved by many, it was also evaluated. In general, when comparing unidirectional UDP vs. TCP results, the ping RTT is lower for the same test case using UDP due to most of the data being in one direction.

When testing in the upstream with TCP, as previously discussed, the only indication we have of RTT, because of using Microsoft Windows clients with iperf3, is the ping RTT statistics. When testing the same non-rate limited scenario in the upstream direction, we do see improvements with OFDMA enabled on TCP and UDP.

Shown in Figure 5 is UDP in the upstream direction at UDP Payload of 1460 bytes, the average RTT of a ping is reduced by more than half for OFDMA enabled scenarios in the maxed-out channel test. For example, with all Wi-Fi 6 clients used and OFDMA disabled, the average RTT of a ping was 14.41 ms to the client under test (CUT). With OFDMA enabled on the AP for the same four Wi-Fi 6 client test, the average ping to the CUT was reduced to 6.76 ms. Even better results are seen with UDP payloads of 536 bytes in this same test scenario, however, the throughput is better with the UDP payload of 1460 bytes.

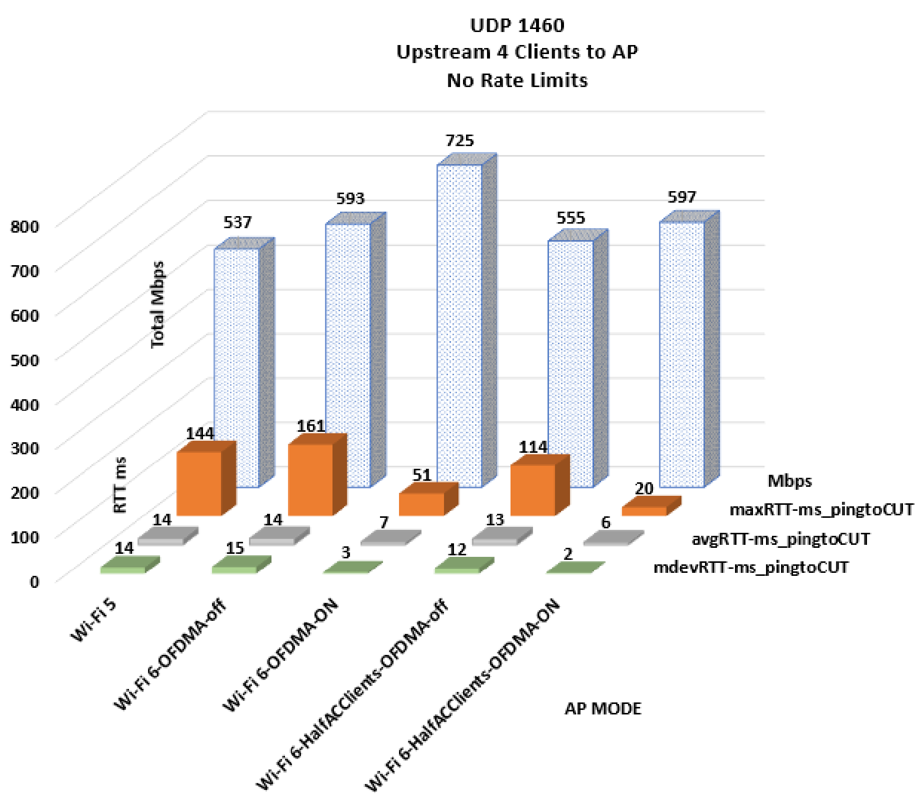

Figure 5 – UDP Upstream No Rate Limits

In this completely saturated airtime scenario, the per ping sample to the CUT was graphed in Figure 6; it is quite revealing in the OFDMA disabled to OFDMA enabled comparison for upstream traffic in a completely used airtime scenario. The AP scheduling upstream transmissions allows for very quick and continuous access to the channel. In Figure 6 below, a much tighter and lower latency can be seen with the blue line with OFDMA enabled and is a visual representation of the median deviation of RTT latency changing from 15 ms to 3 ms which was revealed in the previous Figure 5 as well as the max RTT recorded reducing 3 fold.

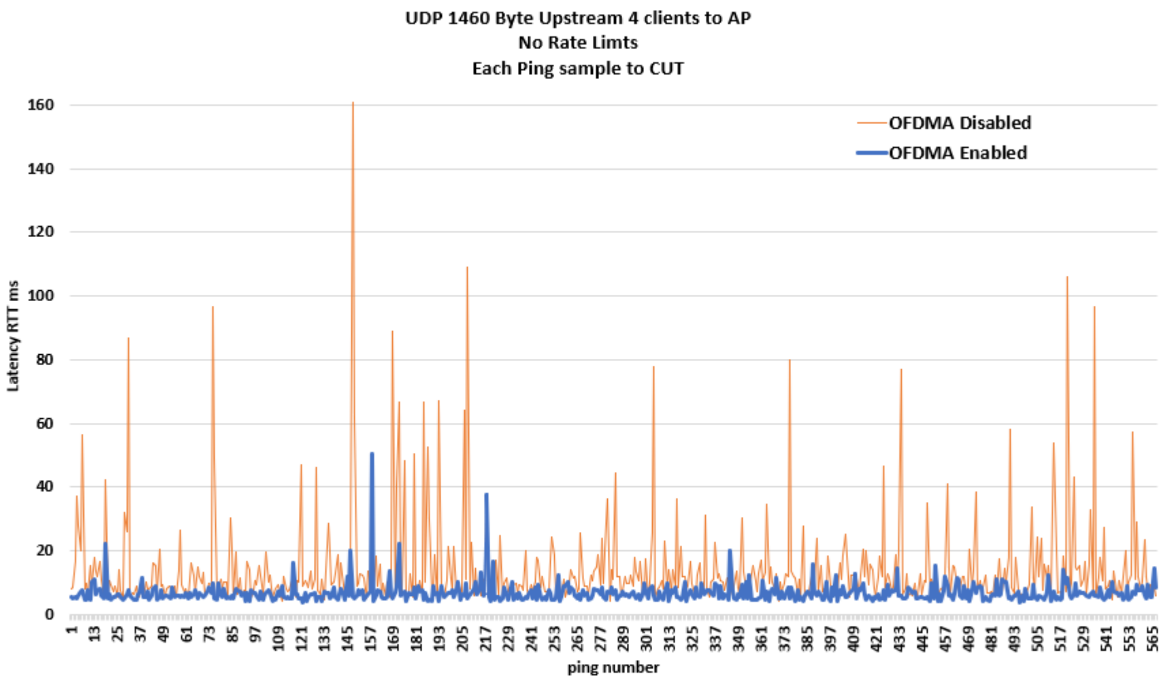

Figure 6 – Per Ping Sample OFDMA Disabled vs. Enabled

In Figure 7 below, the same scenario of UDP and maxed out upstream traffic is compared between AP set to Wi-Fi 5 (gray line) vs. OFDMA enabled (blue line) and disabled (orange line) while half the clients are set to Wi-Fi 5 only mode. This test mode was tried in all scenarios to evaluate if benefits to Wi-Fi 6 clients can be realized in multi-client scenarios that include previous generation Wi-Fi 5 clients. The latency reduction benefit is still seen to the Wi-Fi 6 client under test with half the clients being unable to use OFDMA but still sending traffic upstream.

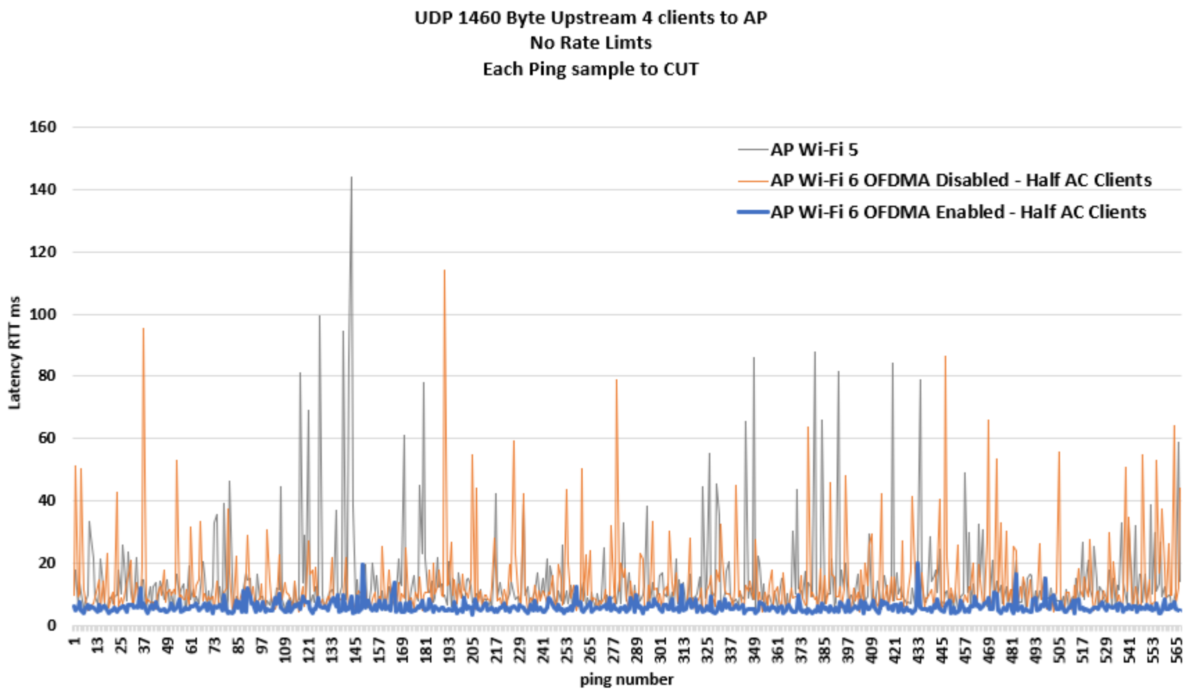

Figure 7 – Per Ping Sample Wi-Fi 5 vs. OFDMA Disabled/Enabled with half Wi-Fi 5 Clients

Next is an example of downstream TCP traffic while maxing out the utilization of the channel with no rate limits being used. There was a particular improvement seen in the mixed client population test scenario where half of the network's clients, are set to Wi-Fi 5 mode only. The other two clients, including the CUT, are left to support Wi-Fi 6 and OFDMA. In the following scenario TCP was used in the downlink direction at 1460 byte MSS with no rate limit set on iperf3 while measuring the client under test's RTT latency. This scenario again shows a benefit can still be seen on a Wi-Fi 6 client under test while half the clients in the test are Wi-Fi 5 clients and don't have an ability to use OFDMA. The AP can schedule the OFDMA clients together, in this case 484 RU each in a group while Wi-Fi 5 clients are served in alternating fashion with the OFDMA group of size 2 as transmit opportunities are won.

Shown in Figure 8, the maximum RTT latency (orange bars) recorded during a 60 second test showed a latency reduction of about 50% with OFDMA enabled and was seen in both max ping RTT as well as max data RTT (blue bars) to the Wi-Fi 6 client under test. The average RTT (gray bars) only showed slight improvements in OFDMA enabled case, however the max RTT recorded was meaningfully lower. This reduction in the max RTT recorded is very likely to improve the QoE for the end user in the most demanding maxed out scenarios of fully used channel airtime from mostly downstream traffic. The mean deviation of the latency (green bars) or jitter as reported from the approximately 567 pings in the 1 minute test time showed a reduction to just 8.42 ms with OFDMA enabled while OFDMA disable was at 22.26 ms and legacy Wi-Fi 5 was at 50.98 ms.

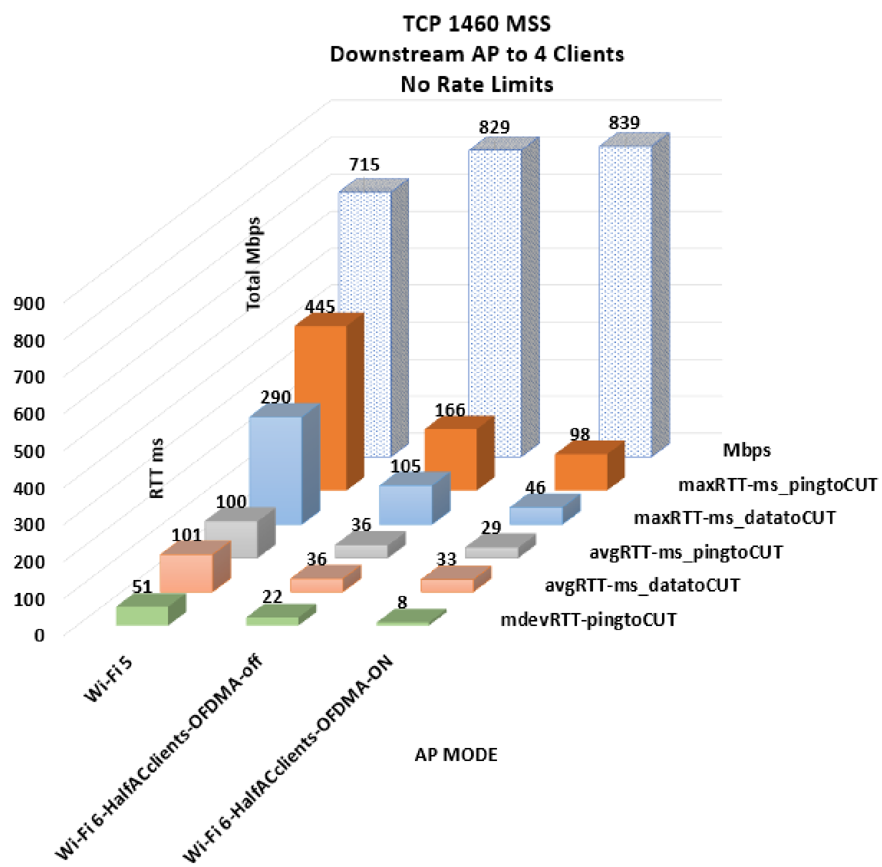

Figure 8 – TCP Downstream No Rate Limits

The same scenario of the AP at Wi-Fi 6 but half the clients being set to Wi-Fi 5 is graphed below in Figure 9 on a per ping basis to the CUT and shows a good improvement in reduction of frequency of RTT latency spikes with OFDMA enabled (blue line) vs. OFDMA disabled (orange line). The AP set to Wi-Fi 5 mode (gray line) is shown as well to show how variable Wi-Fi 5 networks can be when loaded.

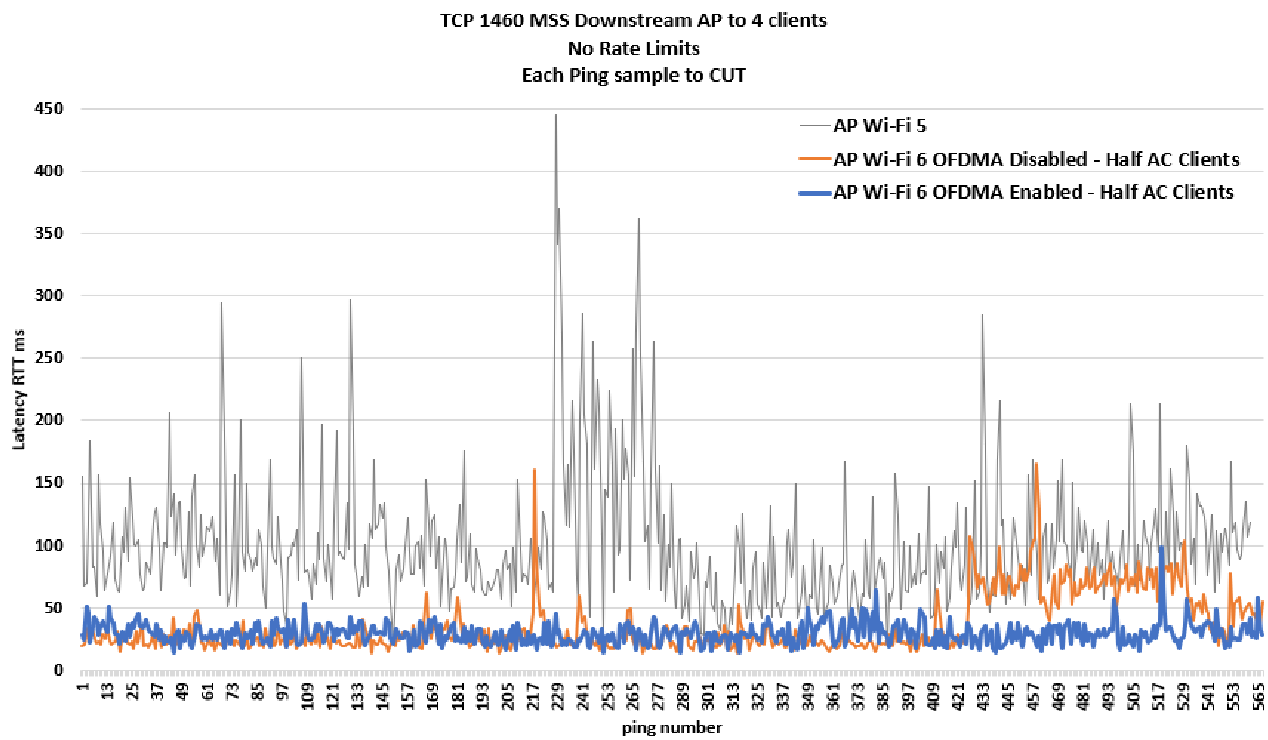

Figure 9 – Per Ping Sample Wi-Fi 5 vs. OFDMA Disabled/Enabled with half Wi-Fi 5 Clients

In summary, for the non-rate limited scenarios with heavy in-network channel utilization, the OFDMA enabled tests showed improvements in average RTT and max RTT as well as mean deviation of RTT latency. For upstream traffic utilization UDP traffic tests were the easiest to see drastic improvements while for the TCP traffic tests the most latency reduction was seen in downstream direction test cases. In most non-rate limited test scenarios there was an improvement for Wi-Fi 6 client latency even when half the clients receiving background traffic were Wi-Fi 5 clients.

8.2. Rate-Limited Test Scenarios with Channel Utilization Set to 90%

In this section's test scenarios, the bitrate limits for each of the other 3 clients were set to a limit of 90% of what each had achieved without rate-limits in baseline testing, per AP mode. This caused approximately 90% channel utilization as compared to the baseline unlimited test case. Two of this scenario's results were selected to discuss below.

In the following scenario with results shown in Figure 10, the client under test was set to send 50 Mbps upstream with the channel utilization at approximately 90% with upstream traffic from the other clients. This scenario represents a very highly utilized channel from the other clients while a moderate throughput demand in the upstream is created from the CUT. The OFDMA enabled test, shows a reduction in the max RTT ping (orange bars) from 313 ms to 121 ms to the CUT with an average RTT ping (gray bars) that reduced from 50 ms to 37 ms. The median deviation or jitter (green bars), also decreased from 44 ms to 21 ms for the same OFDMA enabled vs. disabled comparison to the CUT. Additionally, the data in Figure 10 records a decrease in the max RTT ping seen when OFDMA is enabled with half the clients set to Wi-Fi 5 mode; this shows some improvement can still be realized in mixed client populations of Wi-Fi 5 and Wi-Fi 6 clients.

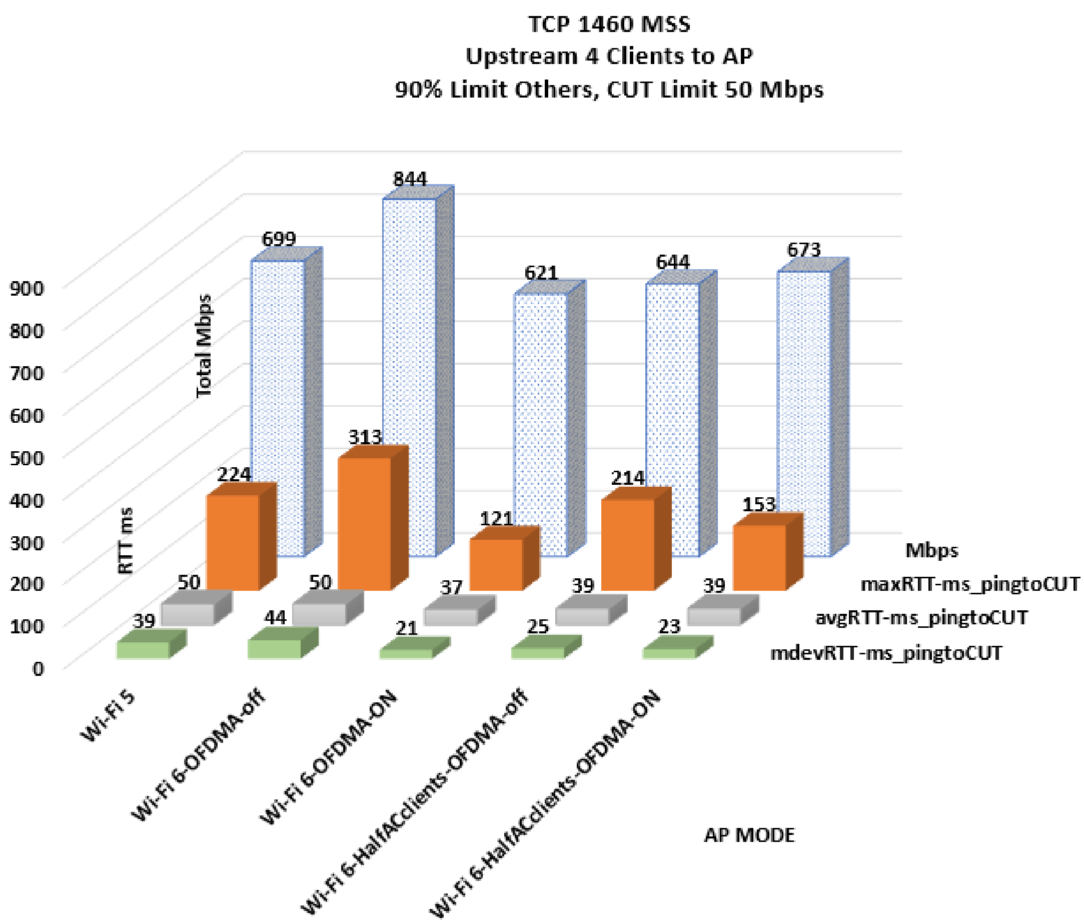

Figure 10 – TCP Upstream, 90% Limit Others, CUT Limit 50 Mbps

In Figure 11 below, the per ping sample between OFDMA disabled (orange line) and enabled (blue line) is graphed for the same scenario depicted in Figure 10 and shows visually the drastic improvement in the max RTT latency observed with much lower spikes in latency and a much tighter jitter or median deviation of the latency. The median deviation was shown previously to have reduced from 44 to 21 ms from OFDMA disabled to enabled and is seen below with a smaller range of values during the test, meaning the blue line's spikes in latency were much lower with OFDMA enabled.

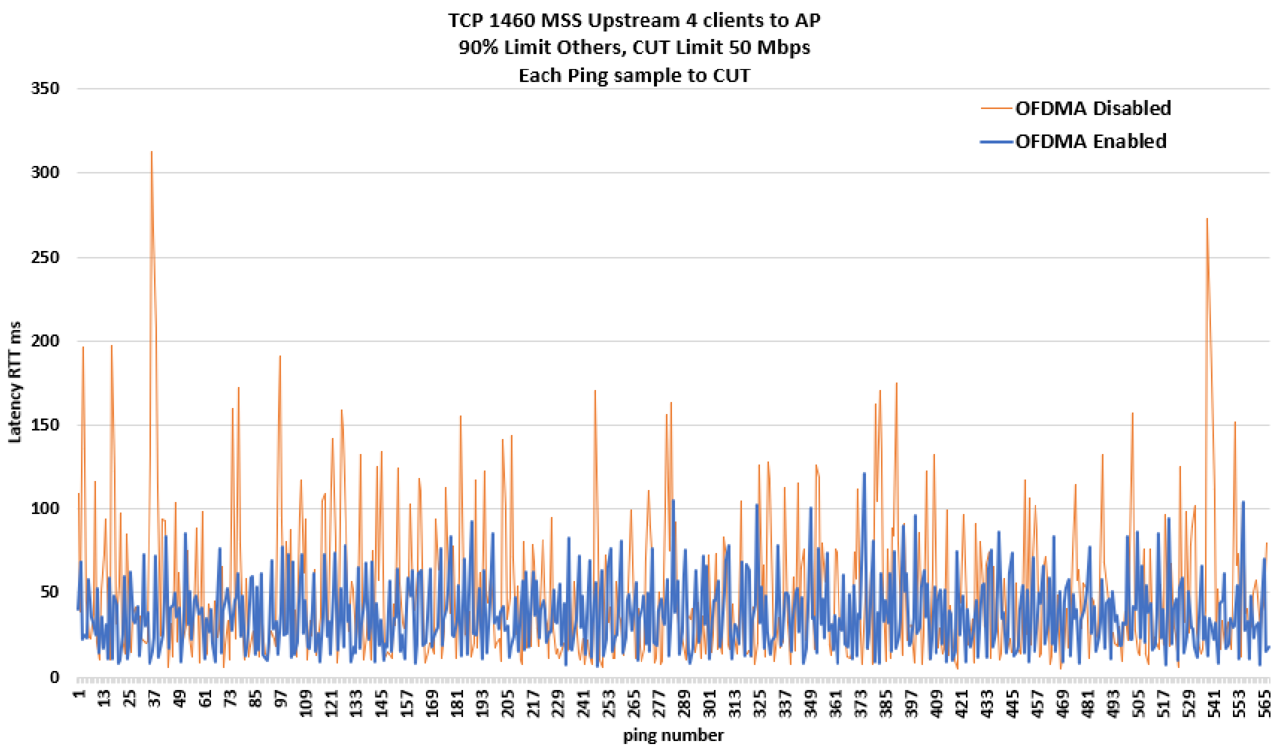

Figure 11 – Per Ping Sample OFDMA Disabled vs. Enabled

The next scenario, for an approximately 90% utilized channel, was with downstream TCP traffic at a smaller 536 byte MSS and was more compelling on a different AP shown below in Figure 12. Other clients were limited to approximately 90% and the rate-limit of the CUT was set to 500 Mbps. This test scenario is like the maxed-out scenario in that the total allowed rates are well over the ability of the channel to support but serves as a test to see if the CUT can get more throughput in the channel with less latency on that data when OFDMA is enabled. Indeed, in addition to improving the throughput achieved, latency was also reduced for the CUT. The CUT throughput (dark blue bars) seen in this scenario for each AP mode included Wi-Fi 5 achieving 69 Mbps, Wi-Fi 6 with OFDMA disabled achieving 153 Mbps, and Wi-Fi 6 with OFDMA enabled achieving 173 Mbps. The scheduler preferred to give more equal access to the channel preserving lower latencies for all clients while still allowing more traffic through on average to the CUT in the OFDMA enabled test. This contrasts with assigning a higher number of tones for this client and others in two groups or just 996 tones to the CUT to try to allow the higher TCP limit of 500 Mbps, however it did not do this. In Figure 12 below, both the iperf3 data max RTT (light blue solid bar) and iperf3 average RTT (orange bars) have reduced by about 30 - 40%.

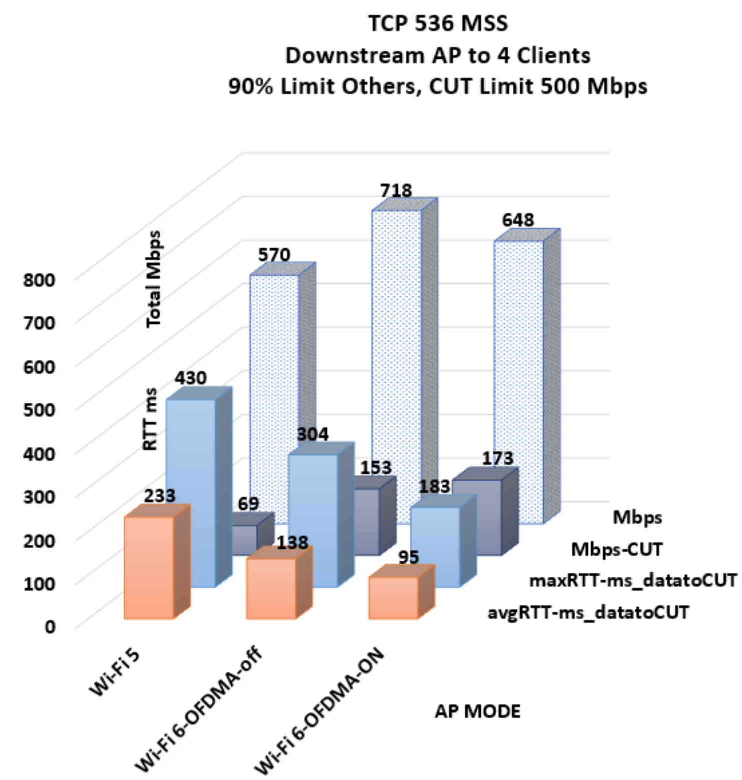

Figure 12 – TCP Downstream, 90% Limit Others, CUT Limit 500 Mbps – AP 2

In summary, the approximately 90% utilized channel scenarios showed most improvements in TCP upstream and UDP upstream scenarios. With UDP downstream scenarios the results were usually worse at all CUT bitrates tested and may allude to the fact that the scheduling overhead is just too much to see a benefit over OFDMA disabled test cases. In UDP downstream tests, low latency is already able to be achieved to begin with since the AP is already in control of scheduling traffic one at a time to each client without much interruption in the other direction since only layer 2 acknowledgements need to be received. With downstream TCP scenarios the lower utilization tests of the CUT were not able to realize much of an improvement, however, with slightly higher bitrates to the CUT such as 50 Mbps or 500 Mbps, some moderate improvements were measured.

8.3. Rate-Limited Test Scenarios with Channel Utilization Set to 50%

In this section's test scenarios, the bitrate limits for each of the other 3 clients were set to a limit of 50% of what each had achieved without rate-limits in baseline testing, per AP mode. This caused approximately 50% channel utilization as compared to the baseline unlimited test case. Two of this scenario's results were selected to discuss. As channel utilization is reduced from 90% to 50% the RTT latency benefits are seen to less of an extent but are still noticed if the CUT has a high bitrate.

In the following scenario, the CUT is set to receive at a 500 Mbps limit while the other clients are set to a limit of 50% to represent approximately 50% channel utilization before the CUT's traffic. The 500 Mbps rate limit set on the CUT is a higher demand or throughput limit than available unused channel airtime and higher than what can be achieved in only a 242 tone RU alone. This tests if the scheduler will give higher RU allocation, such as 484 tones, to this client under test to achieve its higher throughput needs. Or, if the scheduler would cause throughput degradation to the CUT by assigning a smaller than required RU allocation such as 242 tone to maintain a single group of all 242 tone RUs to each of the four Wi-Fi 6 clients, in the OFDMA enabled test case.

Since the client throughput (dark blue bars) can achieve a rate over time that is higher than the rate possible in just a 242 tone RU, this scheduler sacrificed some latency reduction potential that could have been obtained for all clients, at the expense of creating multiple groups with higher number of tones for each client. In this way, a higher throughput on a client demanding more than the others was accommodated. Therefore, the latency reductions shown in Figure 13 are more muted, and a great example of scenarios that are not expected to show a big RTT latency reduction because of the test conditions and scheduler deciding to allow as much of the CUT's higher throughput traffic as possible.

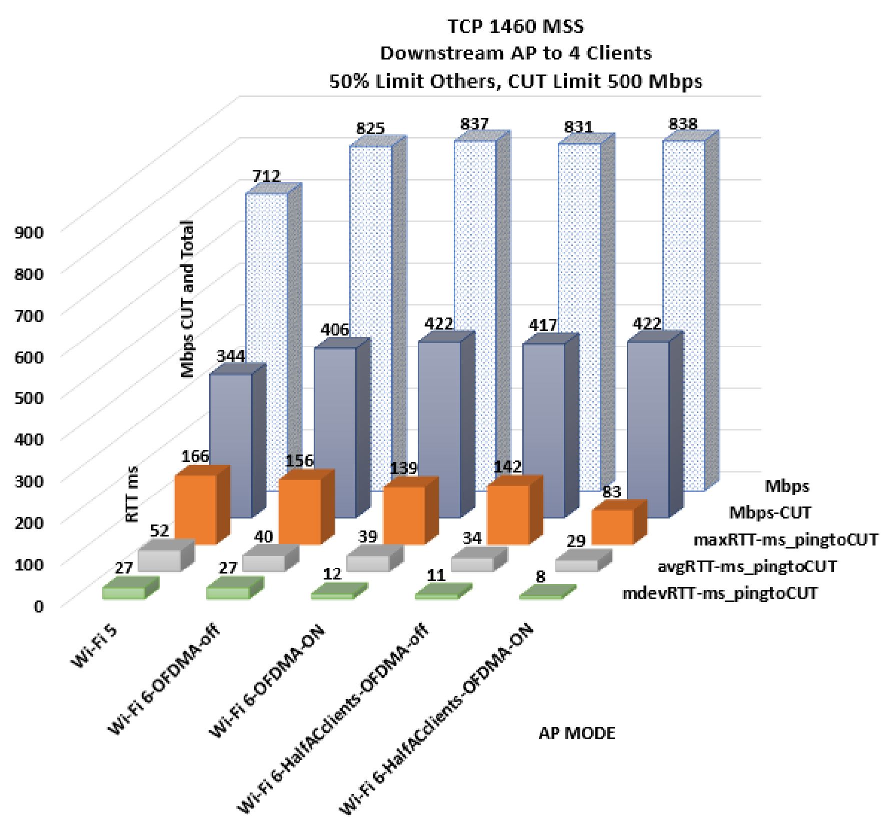

Figure 13 – TCP Downstream, 50% Limit Others, CUT Limit 500 Mbps

In another scenario of approximately 50% channel utilization allowed on other clients, downstream UDP traffic is used with a moderate limit of 50 Mbps to the CUT. The average RTT latency (gray bars) on the CUT is about 1 to 2 ms higher with OFDMA enabled. However, the max RTT latency (orange bars) did fall from 104 ms to 39 ms with OFDMA enabled, and from 86 ms to 54 ms in the mixed Wi-Fi 5 client tests with OFDMA enabled. This is an example of a scenario that may not show great average latency reduction on the Wi-Fi 6 CUT but does show a lower max RTT latency and would result in a better QoE for the end user. These results are shown below in Figure 14.

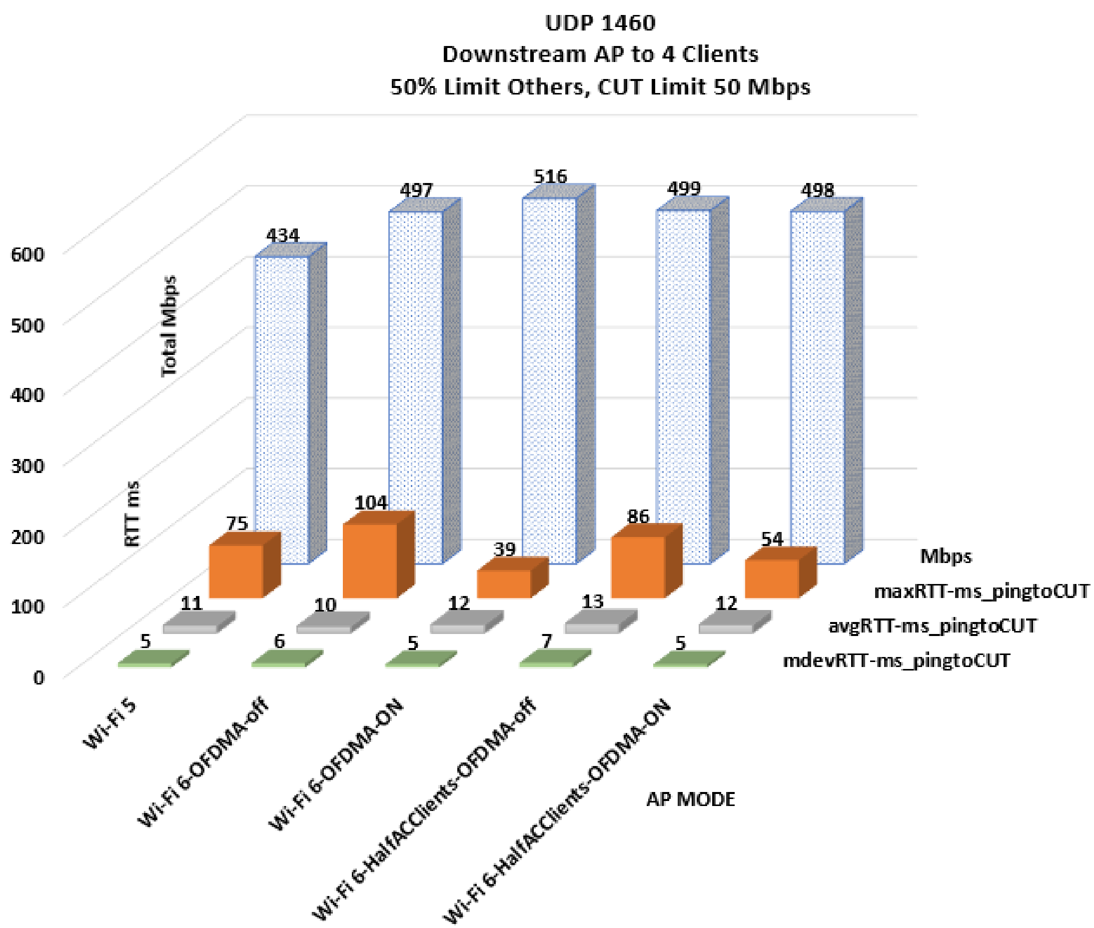

Figure 14 – UDP Downstream, 50% Limit Others, CUT Limit 50 Mbps

To illustrate the moderate benefit of a lower max RTT for this scenario, a per ping sample graph in Figure 15 is shown below. The spikes in latency are still seen but not as high, and the slight increase in average RTT ping is also able to be seen.

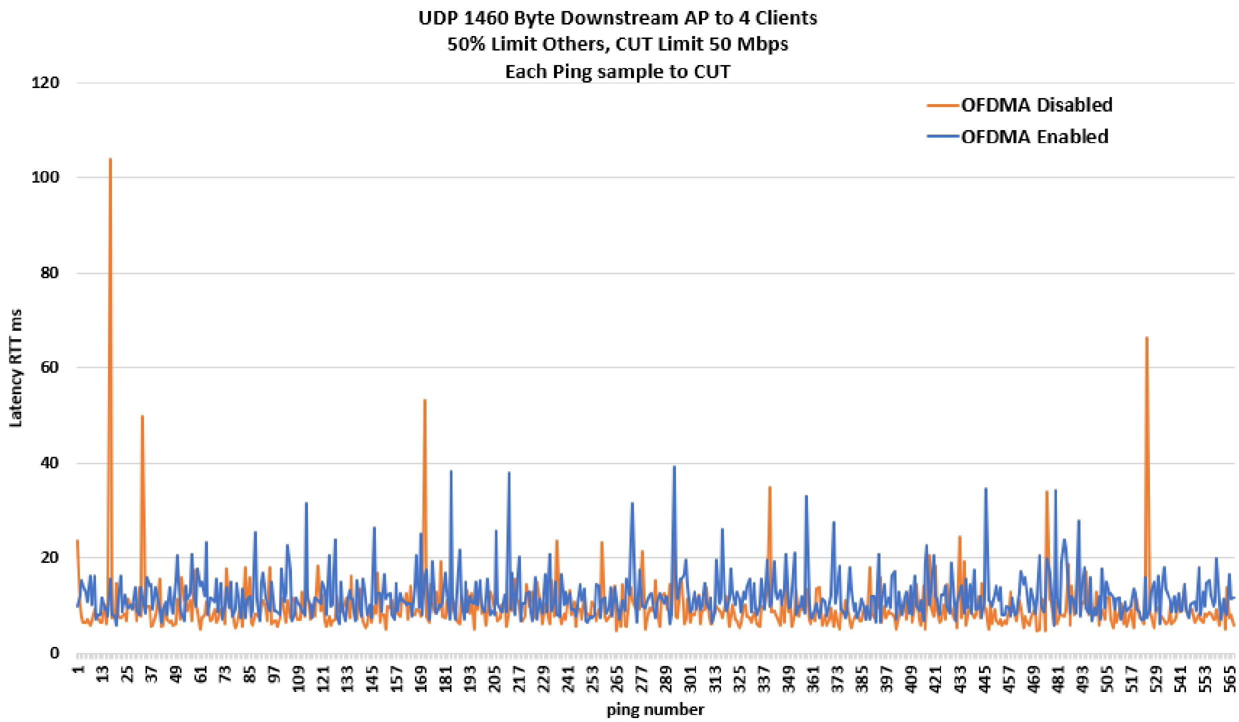

Figure 15 – Per Ping Sample OFDMA Disabled vs. Enabled

In summary, the approximately 50% utilized channel scenarios showed most improvements with OFDMA enabled when the CUT was set to a higher throughput limit such as 500 Mbps. This was certainly the case with UDP in the upstream direction, but not as convincingly seen in the downstream direction. UDP upstream isn't really a use case seen at this high bitrate in real-life and therefore focus was to show UDP and TCP downstream scenarios described above. UDP downstream showed a small improvement in RTT latency with the CUT receiving 50 Mbps. For TCP traffic tests, the only cases improving with approximately 50% channel utilization were in the downstream direction and at 500 Mbps to the CUT. The TCP upstream tests, for the 50% channel utilization scenarios, for the most part showed slightly worse latency.

8.4. Rate-Limited Test Scenarios with Channel Utilization Set to 10%

In this section's test scenarios, the bitrate limits for each of the other 3 clients were set to a limit of 10% of what each had achieved without rate-limits in baseline testing, per AP mode. This caused approximately 10% channel utilization as compared to the baseline unlimited test case. Two of this scenario's results were selected to discuss including one to show the difference between two different AP's in the same scenario.

In the least utilized channel test scenario with other clients set to 10% rate limits, there were less situations found with meaningful improvements in latency for the CUT. One of the only scenarios that showed a compelling difference for a low utilized channel and highly utilized client was in the mixed client mode test with and without OFDMA enabled. In the scenario shown in Figure 16, TCP is used with a 1460 byte MSS and a 10% utilization limit for the other clients while the client under test is set to a rate limited 500 Mbps in the upstream direction. This represents a very high throughput demand in the upstream while the channel utilization is quite low in the same direction on the other clients. The mixed client mode cases of half Wi-Fi 5 clients and half Wi-Fi 6 clients showed a compelling difference in the latency experienced on the CUT during this high throughput of a lightly utilized channel in the upstream. The CUT was able to achieve the 500 Mbps upstream in both cases, yet the RTT max latency (orange bars) as well as the RTT average latency (gray bars) both were reduced approximately 3-fold.

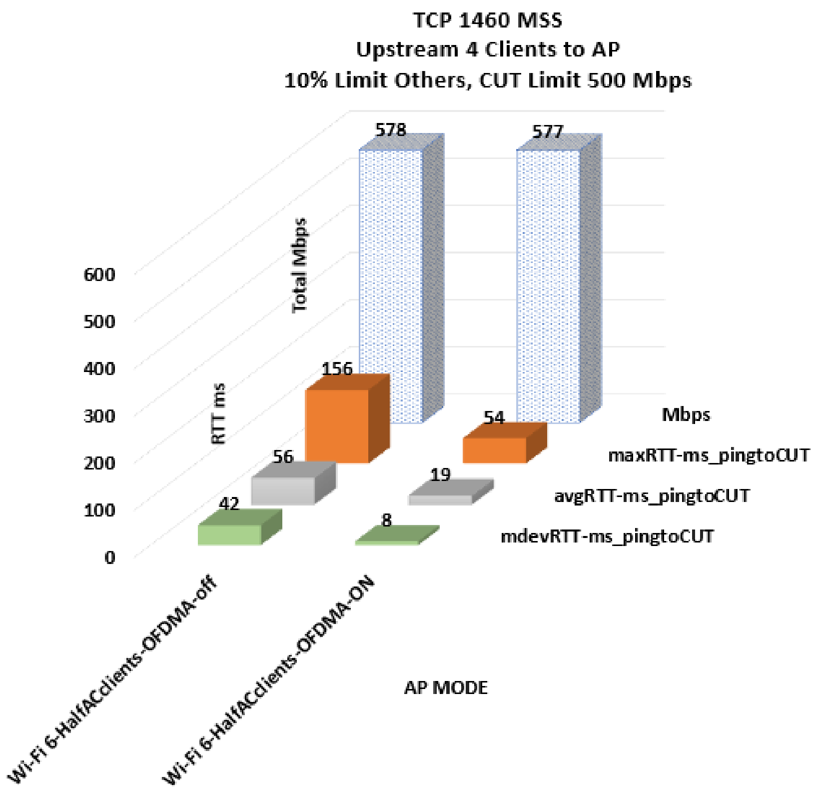

Figure 16 – TCP Upstream, 10% Limit Others, CUT Limit 500 Mbps

The same scenario is graphed with per ping RTT latency data in Figure 17 below and really illustrates the 5 times lower RTT median deviation in the ping data to the CUT with OFDMA enabled (blue line) in the mixed half Wi-Fi 5 client test. The data set to flow upstream from the CUT was able to achieve 500 Mbps in both scenarios, but the OFDMA enabled case did it with a much lower average RTT latency as well as considerably less jitter.

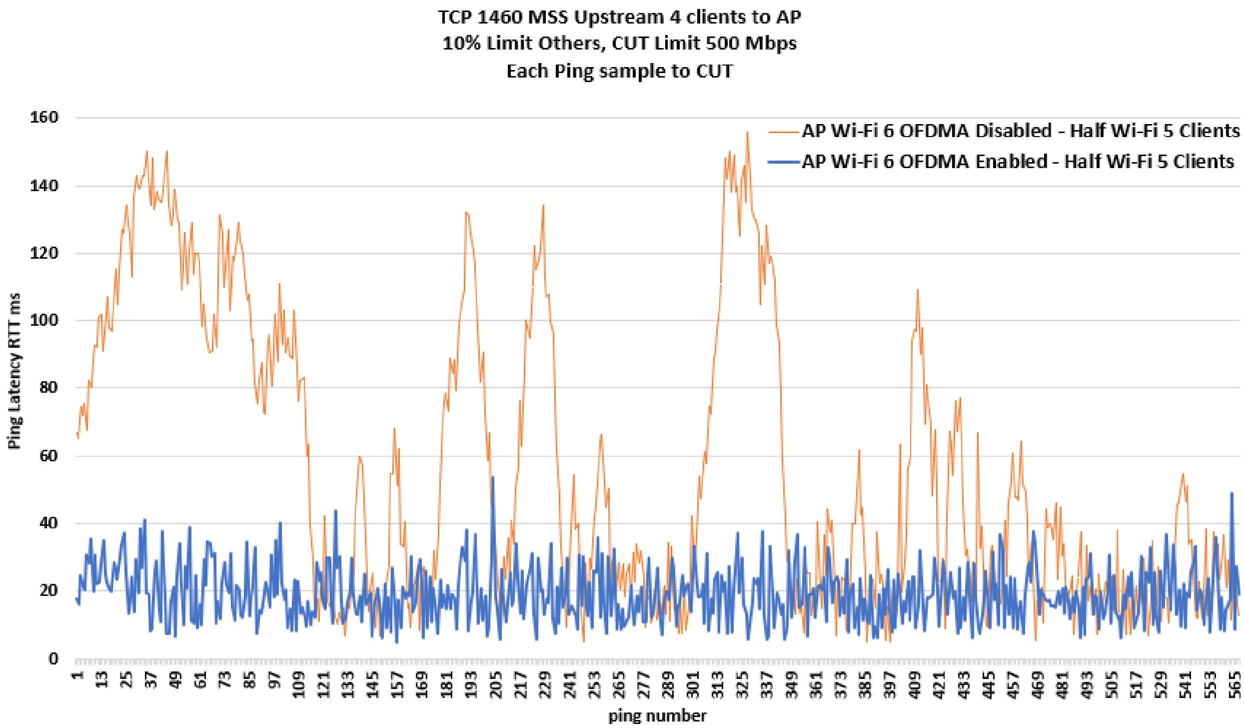

Figure 17 – Per Ping Sample OFDMA Disabled vs. Enabled with half Wi-Fi 5 Clients

As mentioned before, not all scenarios showed reduced latency and many also show increased latency, especially with lower utilized channels or lower rates to the CUT. Most notably, differences between different APs are very easily seen when compared in the same test conditions and scenarios. The same scenario with two different APs will illustrate this extreme difference that can be seen.

The scenario compared here was TCP 1460 byte MSS with 10% traffic in the downstream direction with the CUT receiving a rate-limited 5 Mbps. So, this represents lightly used channel conditions and a low throughput on the CUT. This is actually a very common scenario and may represent the most often situation experienced in the home.

The first AP's results, in Figure 18, show that with OFDMA enabled or disabled the same average RTT latency (solid light blue bars) for the iperf3 data flows is seen at 4 ms. The max RTT latency with OFDMA enabled increased from 5 ms to 7 ms and reduced max RTT latency from 7 ms to 5 ms in the mixed Wi-Fi 5 client mode case. All these latency values, both average and max RTT are acceptably low for the local LAN segment for most applications but are shown here to illustrate the improvements are not usually seen in low to medium use cases with 4 clients. Also, this simple result is given as a contrast to the result with the same scenario on a different AP.

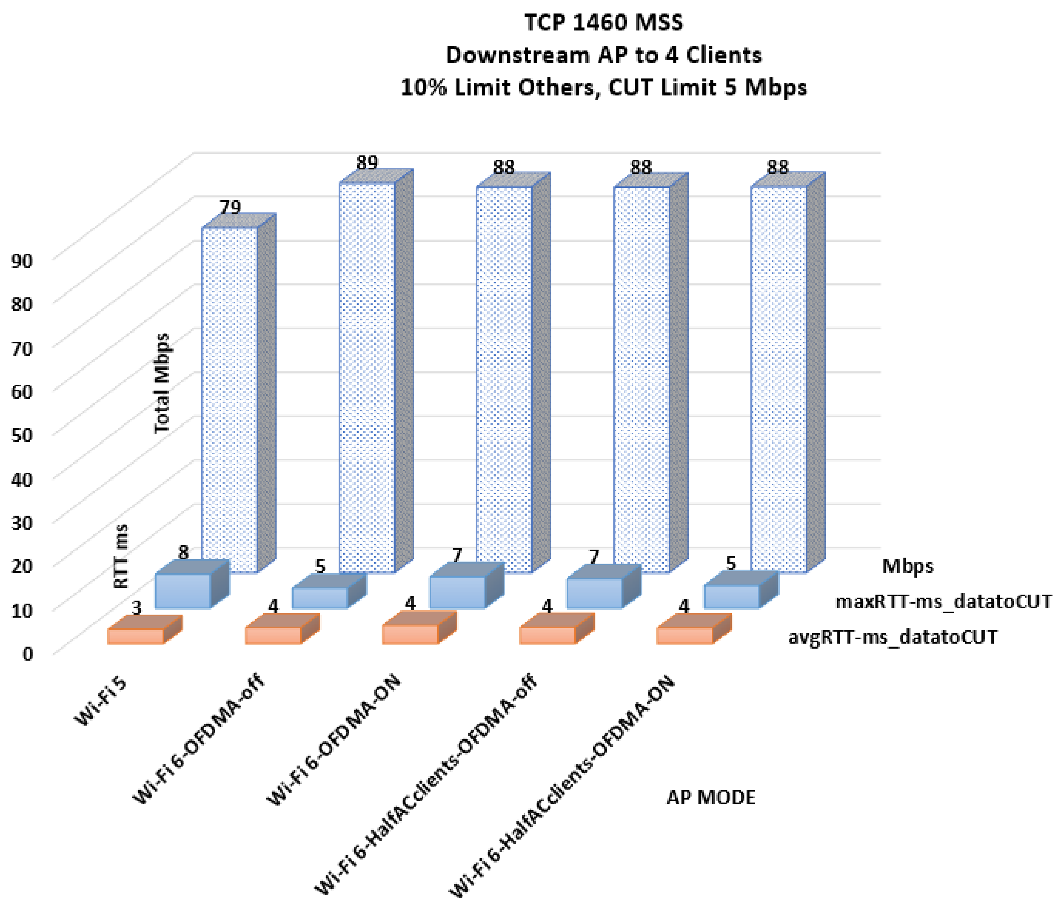

Figure 18 – TCP Downstream, 10% Limit Others, CUT Limit 5 Mbps – AP 1

In Figure 19, the same scenario is tested using a different AP and revealed some issue with increased latency for the OFDMA enabled test cases with all Wi-Fi 6 clients and again in the mixed half Wi-Fi 5 client mode test as well. This is likely from scheduler problems with deciding if the clients should be in a group or not and the overhead associated with creating and tearing down groups when low bitrates are used. The scheduler in this AP should be improved to create groups early, even with low bitrates, to receive upstream TCP acknowledgements coming back as data frames in OFDMA transmissions.

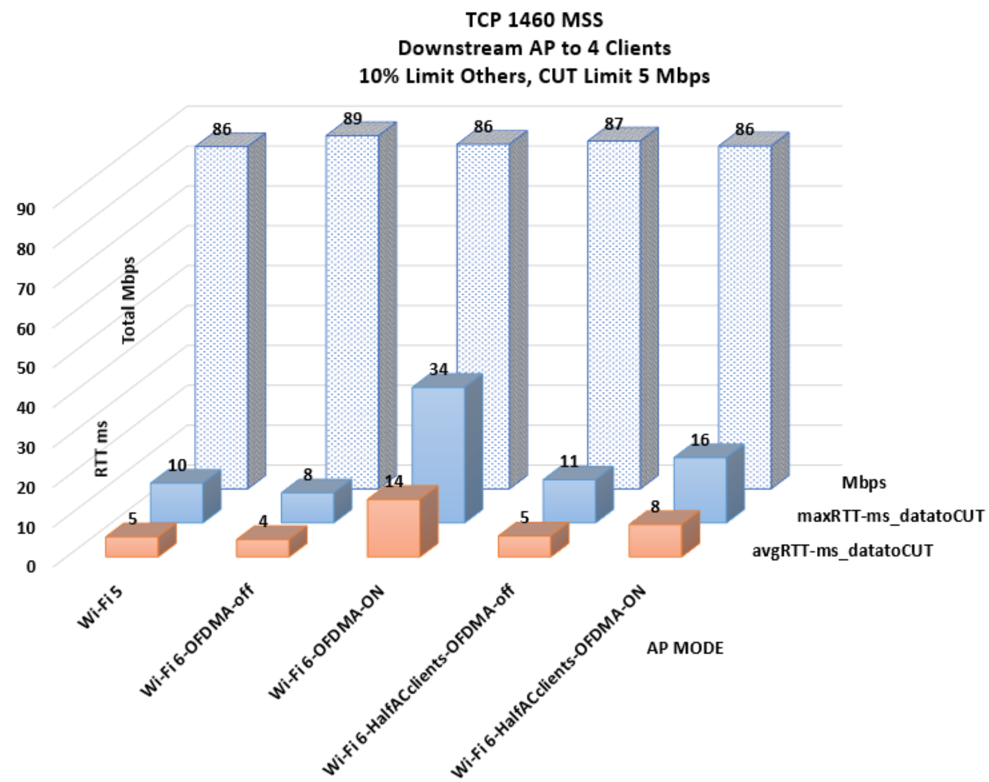

Figure 19 – TCP Downstream, 10% Limit Others, CUT Limit 5 Mbps – AP 2

An area of further testing to be completed is to investigate at what bitrate threshold AP schedulers are considering adding clients to OFDMA groups. A client at distance could be achieving a low bitrate but are using a lower MCS rate and airtime usage would still be high; these low bitrate and high airtime clients should not be denied scheduling into OFDMA groups. A threshold based on predicted airtime usage would be a better approach for such situations when clients are not close to the AP and low MCS rates with high airtime usage are being used. The data shows that highly utilized channels can benefit greatly by OFDMA and highly utilized refers to airtime usage, not bitrates. Therefore, it is likely beneficial that chipsets use thresholds that are airtime based, not throughput based if the scheduler requires such thresholds.

In summary, the approximately 10% utilized channel scenarios showed some small improvements with OFDMA enabled when the CUT was set to 5 Mbps and TCP was used with half Wi-Fi 5 clients in either direction. It was also an easy result to showcase differences in implementation of OFDMA in different chipsets. In other TCP tests, with the bitrate to the CUT set to 50 Mbps or 500 Mbps, latency generally was slightly worse with OFDMA enabled with either direction of traffic. However, an exception was the mixed client scenario with upstream traffic of 500 Mbps to CUT and 10% utilization with other clients, a large reduction in latency was seen with OFDMA enabled and was shown in Figure 16. UDP traffic scenarios with CUT limits of 1 Mbps, 50 Mbps, and 500 Mbps showed slight improvements in the downstream direction tests while the upstream direction only showed an improvement with 500 Mbps from the CUT.

8.5. Rate-Limited Test Scenarios with LAN Hosted Gaming

Online gaming requires quick and responsive connections to have a good QoE. As explained in more detail earlier, a LAN Ethernet hosted CS:GO game was played within the LAN on the CUT while monitoring the RTT Latency reported on the game screen in real-time. Simultaneous to that game play is iperf3 data of 1 Mbps flowing to or from the same CUT and pings just as in prior scenarios. This provides information about the responsiveness from the CUT in the traditional RTT iperf3 data and ping data, in addition to the CS:GO RTT latency reported on the screen overlay.

The 10%, 50%, and 90% usage levels of the other clients as well as iterating over the two packets sizes and AP modes showed similar results to testing without the game play. The scenario that showed an obvious improvement in game play was the heavily utilized 90% TCP upstream traffic scenario, the results are shown in Figure 20. A real-life example of this scenario might include one or more clients in the house backing up data or uploading videos to the cloud while another client is trying to play an online game. Notice the max RTT ping (orange bars) reductions from 212 ms to 69 ms with OFDMA enabled and from 196 ms to 74 ms with half the clients being Wi-Fi 5 clients. The RTT median deviation ping (green bars) also shows a reduction of approximately 50% which alludes to more consistent and lower pings achieved in this scenario to the CUT with OFDMA enabled regardless of half the clients being Wi-Fi 5 clients. The average RTT ping (gray bars) was cut in half when OFDMA was enabled with all Wi-Fi 6 clients.

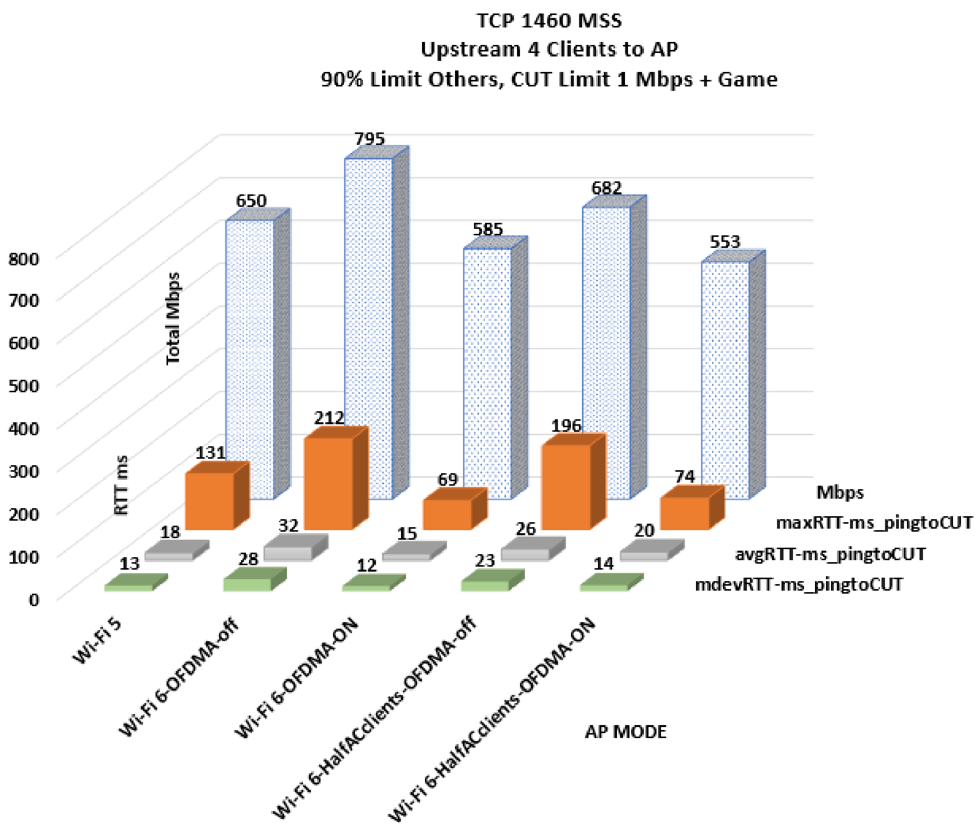

Figure 20 – TCP Upstream, 90% Limit Others, CUT Limit 1 Mbps + Game Play

Consistent with the above measured max and average RTT ping times improving with OFDMA enabled in the heavily utilized channel in the upstream with TCP, the CS:GO game play itself on the CUT also showed improved latency in both OFDMA enabled modes. The average ping recorded during the same sample period of the game play was very close to the RTT Latency reported on the screen overlay during the CS:GO game play itself. This sampling of the CS:GO RTT latency over the 60 seconds is graphed below in Figure 21 and shows the two OFDMA enabled cases performing much better with all Wi-Fi 6 clients (blue line), as well as the mixed half Wi-Fi 5 clients (green line). Both OFDMA enabled tests show an obvious consistent latency or low jitter to the CUT during game play while the channel has very low available airtime with heavy upstream traffic. This was also evident by the reduction in median deviation of RTT pings shown in Figure 20 above.

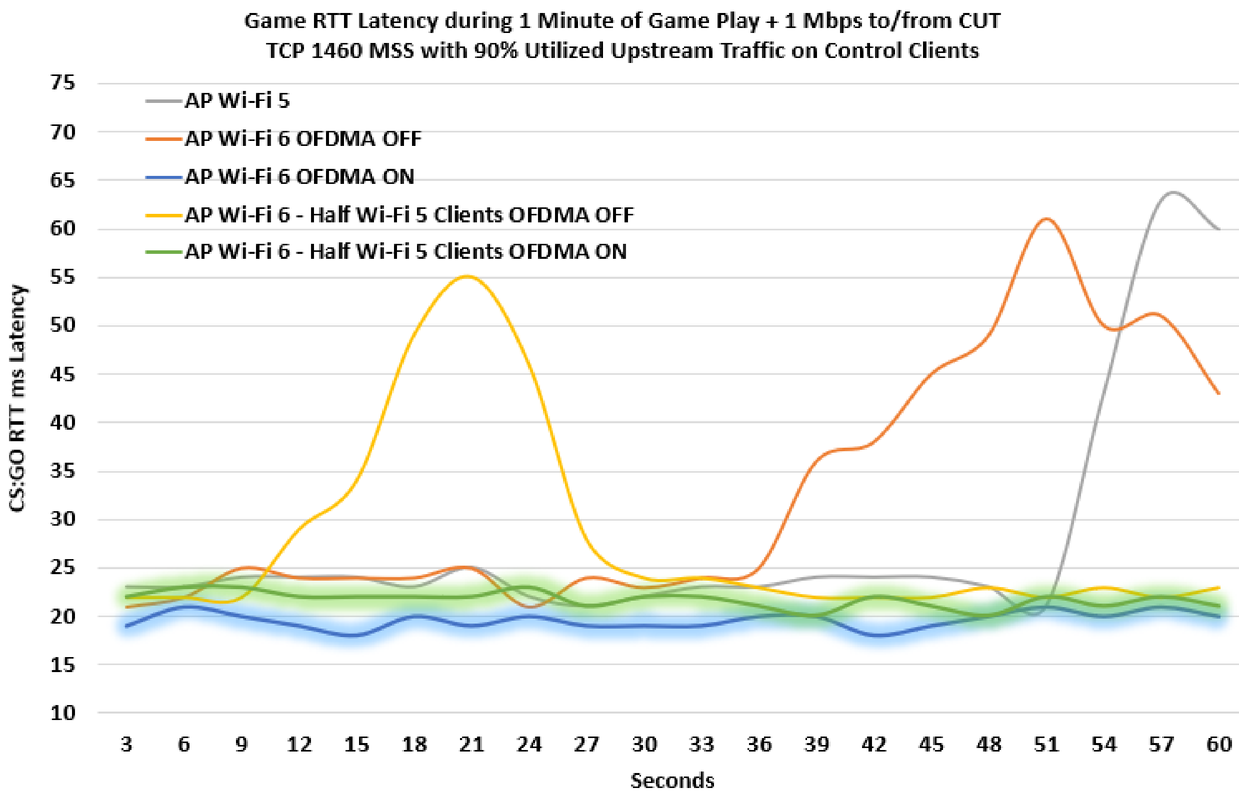

Figure 21 – CS:GO Reported RTT Latency Over 1 Minute of Game Play

In Figure 22 below, the same game play scenario is graphed per ping RTT latency for OFDMA enabled (blue line) vs. OFDMA disabled (orange line). The per ping data shows the same increasing latency for OFDMA disabled at the same points on the graph as were reported by the overlay on screen of CS:GO and depicted in Figure 21 above. In Figure 21 at 36 seconds into the OFDMA disabled test scenario (orange line) latency begins to rise; this correlates with the ping data in Figure 22 for the OFDMA disabled test (orange line) at around the 360th sample which also starts to have more spikes in latency. The ping data samples over time in Figure 22 below show that pings are a good representation of what is also occurring to the actual game flow's latency even if the absolute values are different.

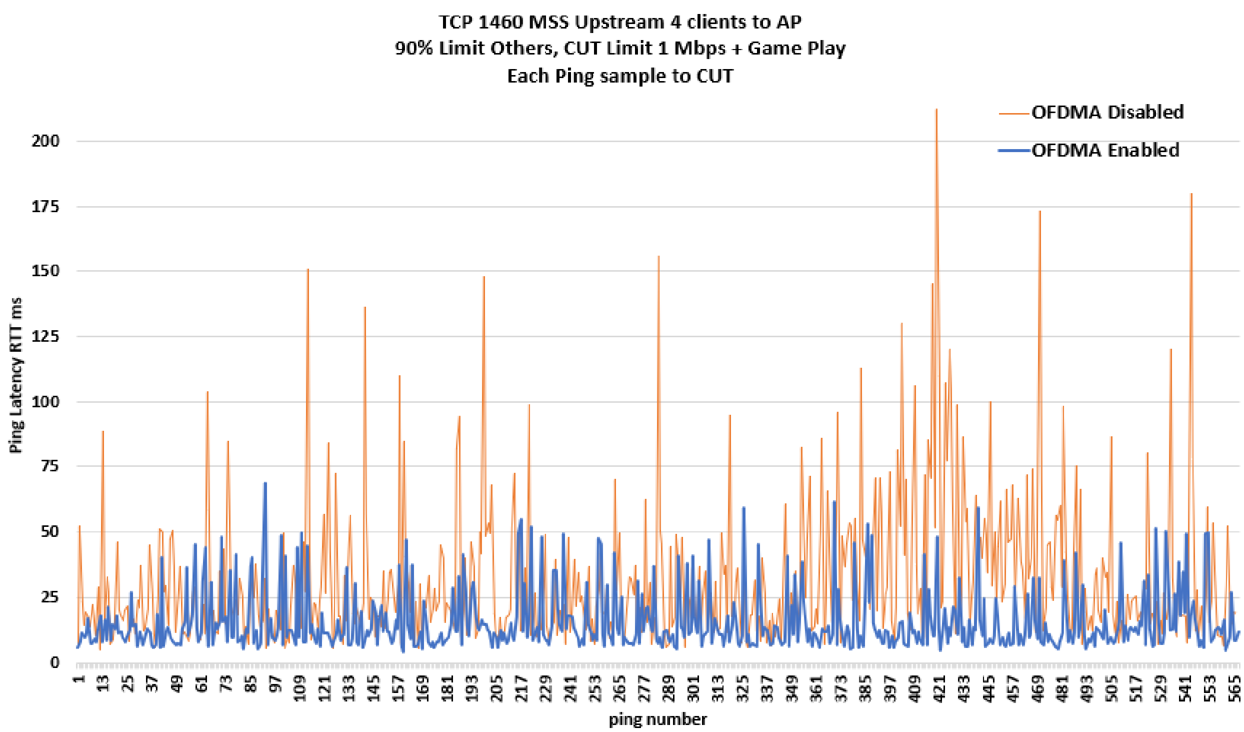

Figure 22 – Per Ping Sample OFDMA Disabled vs. Enabled During CS:GO Game Play

In summary, during online game play via Wi-Fi, OFDMA enabled really improved the variance of latency, or jitter, in highly utilized channels in the upstream for both TCP and UDP. This improvement was measured solely on the LAN as game play was limited to within the LAN. The improvement on the CUT playing the game was realized even with half the clients in the network being Wi-Fi 5 clients. The consistency of RTT latency is much improved with OFDMA enabled in this scenario that represents several other devices uploading data and using approximately 90% of the channel airtime at the same time as a CUT is playing an online game.

8.6. Wi-Fi 6E Testing with Rate-Limited Test Scenarios

Wi-Fi 6E testing revealed some throughput degradation perhaps from range degradation at 6 GHz or from immaturity of firmware on AP or client for 6 GHz. The specific AP used for this testing showed some unexpected degradation in the upstream with OFDMA enabled even in 5 GHz. However, relative differences are still able to be seen by rate-limiting to certain percentages and comparing relative latency during the same approximate channel utilization across AP modes and scenarios.

Like other findings with APs in 5 GHz band, the highly utilized channel of 90% showed the most improvement to the CUT when enabling OFDMA in the 6 GHz band. Figure 23 shows 6 GHz testing with 90% utilization on control clients during TCP 1460 byte MSS testing in the upstream direction with 50 Mbps from the CUT. All stats improved with pings for OFDMA enabled cases, including the median deviation RTT (green bars), average RTT (gray bars), and max RTT (orange bars).

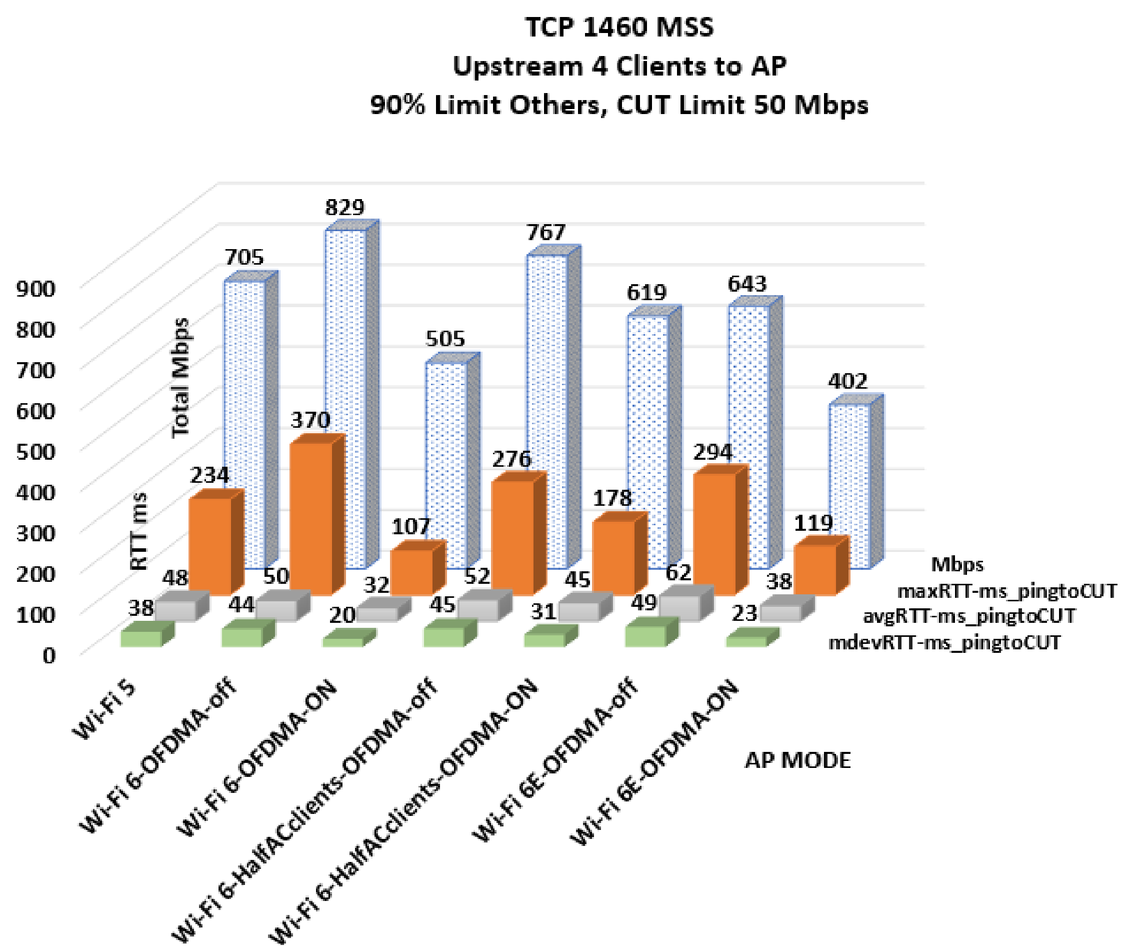

Figure 23 – TCP Upstream, 90% Limit Others, CUT Limit 50 Mbps – AP 3 with Wi-Fi 6E

The same scenario's per ping RTT latency is graphed in Figure 24 below to show Wi-Fi 6E OFDMA disabled (orange line) vs. OFDMA enabled (blue line). The reduction in median deviation of RTT pings as well as the reduction of the max RTT ping recorded is evident with a more consistent and lower average latency for the OFDMA enabled test.

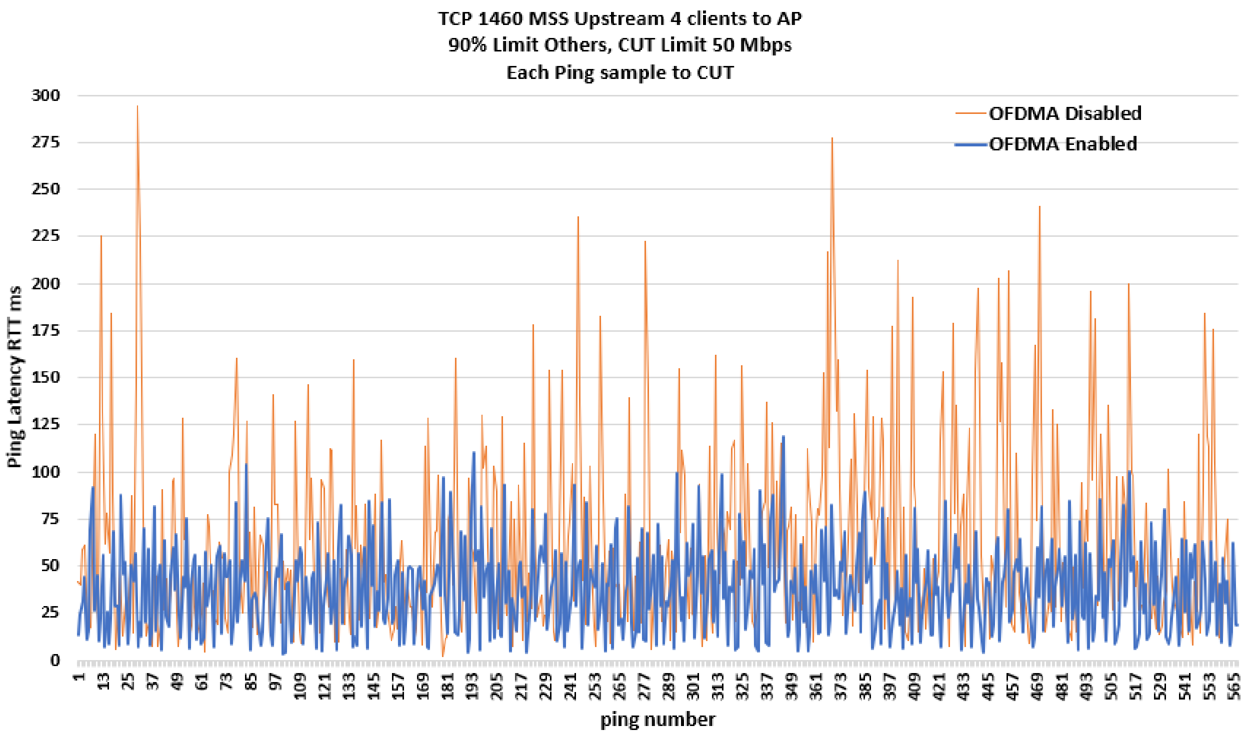

Figure 24 – Per Ping Sample OFDMA Disabled vs. Enabled – AP 3 with Wi-Fi 6E

8.7. Wi-Fi 6E Rate-Limited Test Scenarios with LAN Hosted Gaming

While playing CS:GO on the CUT during the Wi-Fi 6E test scenario, the RTT latency improved the most during upstream 90% channel utilization scenarios while the downstream version of the same test showed about 50% worse RTT latency. Shown below in Figure 25 is the 1460 byte MSS scenario and shows results from a 90% utilized channel in 6 GHz before and after enabling OFDMA. The 536 byte MSS results were similar. This chart is the CS:GO reported RTT latency during a 60 second test while approximately 90% utilization is occurring in the downstream or upstream as labeled by a different colored line. Improvements in latency on a CUT playing a game are not always seen based on if the channel is heavily utilized with downstream traffic or if it is the same percent utilization with upstream traffic. It is also likely something this AP would improve upon with later firmware over time.

In Figure 25, the blue line shows game play latency improved while heavy traffic was flowing in the upstream direction using 90% of the channel. This is to be compared to the orange line showing the same test with OFDMA disabled.

However, the 90% utilization set in the downstream with OFDMA disabled produced a lower average RTT latency shown as the green line in contrast to the OFDMA enabled case shown as the red line, where RTT latency was consistently higher. These results further underscore the point that improvements in latency are not necessarily the same based on the direction of the traffic that is loading the channel. During this game play the downstream traffic load produced lower RTT latency with OFDMA disabled, while the upstream traffic load test produced lower RTT latency with OFDMA enabled.

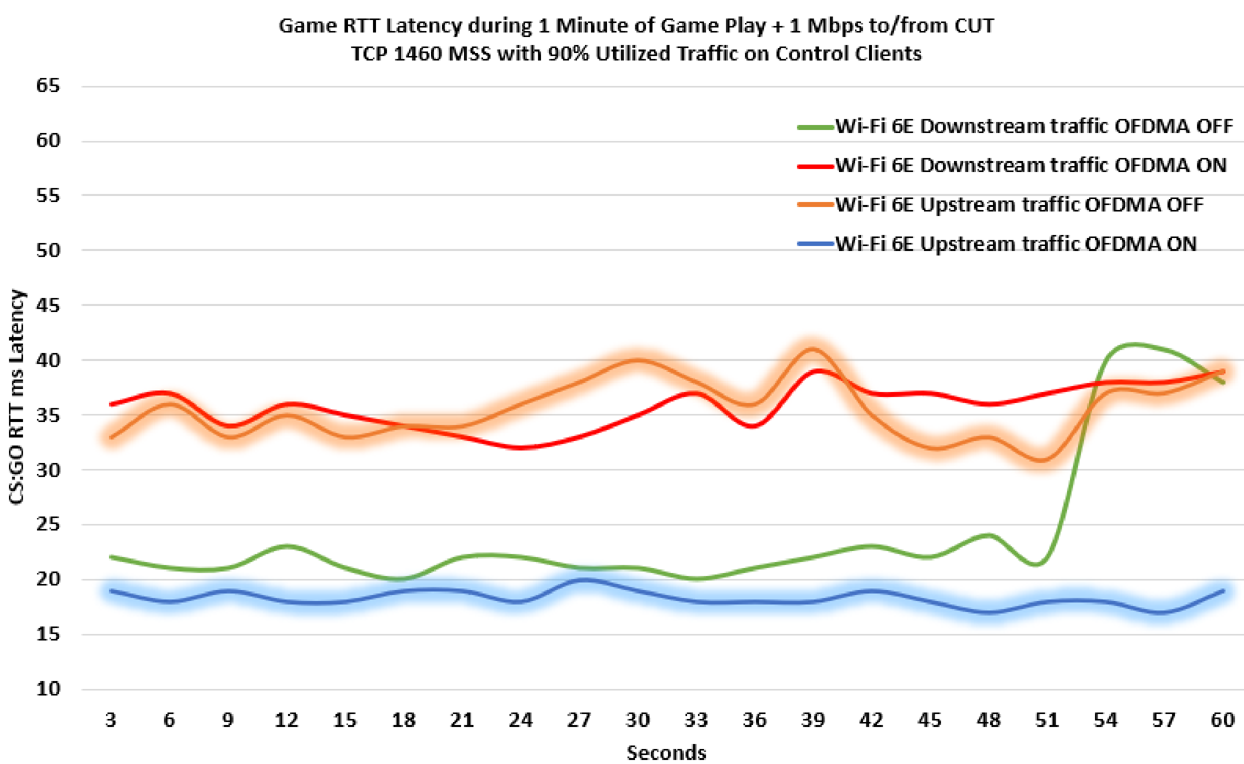

Figure 25 – CS:GO Reported RTT Latency Over 1 Minute of Game Play – Wi-Fi 6E

9. Additional Tests to Consider

There are many opportunities to extend this work and help chipset vendors to improve schedulers for real world scenarios and expectations. A list of additional tests to explore include:

- Create approximate defined percentages of out-of-network traffic on the same primary and secondary channels
- Create approximate defined percentages of out-of-network traffic on only the second half of an 80 MHz channel in the non-primary channels
- Traffic load set on the channel in the opposite direction of the direction of traffic to/from a client under test
- The same test conditions in this paper but with 5, 8, 9, and 13 clients at 80 MHz and 160 MHz to test grouping and usage of mixed RU allocations
- The same test conditions in this paper but with Linux operating system on the clients to get RTT of each TCP data flow in iperf3 in the upstream as well
- 20 clients with 12 clients doing constant 0.1 Mbps traffic to stay utilized while testing other 8 clients with various traffic and channel utilizations
- Characterize how a scheduler allocates RUs as you gradually increase bitrates or airtime percent in use
- Locating clients at near and far locations for 2 groups of power ranges
- Locating clients at near, mid, and far locations for 3 groups of power ranges
- Creating varied but defined traffic load percentages for the other clients to represent low usage and medium usage on many clients in addition to a client under test usage
- Send high data rate to a client while alternating low data rates to another and then vice versa to see if groups are too sticky or if RU allocations are not changed fast enough
- Enable VHT MU-MIMO for evaluation against OFDMA benefits with clients in various locations where VHT MU-MIMO can operate ideally
- Enable HE MU-MIMO for evaluation of scenarios that should use HE MU-MIMO instead of or at the same time as OFDMA – run each separately to determine if schedulers are picking the right mode when both are enabled
- Set up a reliable NTP for better time synchronization between source and clients to get reliable one-way-delay UDP measurements with NUTTCP
- Evaluate the same test case repeatedly to evaluate consistency of results with schedulers for a given test method and given scheduler implementation – is it doing the same thing each time?
- Statically/manually set OFDMA groups and RU number of tones per client to evaluate best case 100% OFDMA performance and compare to results of automatic scheduler operation in throughput and latency
- Use a separate physical Ethernet port for each of the four client's iperf3 traffic
- Use a different Wi-Fi 6/6E client chipset for all the above
- Use one of the many other test tools to evaluate all the above

10. Upgrading to Wi-Fi 6E and enabling OFDMA for Wi-Fi 6

Considering the testing described in this paper, when is the best time to enable OFDMA or deploy Wi-Fi 6E? The answer depends on the needs of the end-user. If the end-user is in a crowded RF environment with a lot of out-of-band Wi-Fi traffic and neighboring APs, upgrading to Wi-Fi 6E now is appropriate. OFDMA alone may not be enough to help overcome persistently crowded channels especially if it is out-of-network traffic on the same channels in use. There are more than enough Wi-Fi 6E devices and laptops available now to take advantage of the extra channels and clean airtime if the end-user needs to solve a problem with airtime availability and can change to a Wi-Fi 6E client or Wi-Fi 6E client adaptor. The clients using Wi-Fi 6E have endured some growing pains with sparse driver update availability and security differences in how clients connect, but these are temporary issues and have already improved greatly over the last year. OFDMA enabled in 6 GHz will continue to improve as chipsets and firmware mature and much of the development and progress on-going in 5 GHz will directly carry over to 6 GHz without much delay.

In environments not overly crowded from out-of-band overlapping 5 GHz traffic, enabling OFDMA on a Wi-Fi 6 AP is a great way to allow what appears to the end user as a more responsive channel while serving the same clients and bandwidth needs in the home. This appears to be the case from the testing completed in this paper, especially if the channel is crowded from in-network traffic. Even mixed client populations of Wi-Fi 5 and Wi-Fi 6 can achieve latency improvements with OFDMA enabled if the Wi-Fi 6 clients are served quicker and simultaneously. Most of the traffic on the internet is TCP, and with TCP traffic the number of situations where latency worsened with OFDMA enabled were about the same as the situations that improved. The situations that didn't improve weren't necessarily situations that needed to improve to have a better QoE to begin with. AP chipset scheduler improvements over time should alleviate many of the issues encountered, because the channel is not contentious in many of the situations in which latency worsened instead of improving. It is yet to be determined if the same benefits observed and described in this paper are also seen with the additional test conditions listed in the Additional Tests to Consider section.

11. Conclusion

This paper sought to define a wide range of test cases to determine which scenarios with today's AP chipsets can achieve reduced latency when OFDMA is enabled. The results showed highly utilized channels stand to benefit the most from enabling OFDMA. Furthermore, higher utilization of the client in use improves the chance of realizing a latency reduction. The upstream tests with OFDMA enabled tended to show more impressive reductions in latency because the clients were not fighting each other for TXOPs in the upstream. The downstream tests did not show the same extent of improvements as were seen in the upstream. In the downstream with OFDMA disabled, the AP is already scheduling transmissions individually as the queues demanded and was less contentious as the AP was in charge in the downstream. VHT and HE MU-MIMO may be more advantageous for latency reductions in the downstream for the environments and schedulers that can use it effectively. The lower bitrates sent to or from the CUT also didn't improve latency as much as the higher bitrates tested and sometimes incurred even higher RTT latency, indicating some chipset thresholds to cause OFDMA scheduling of the CUT were not met.

Test scenarios with 10% utilization from other clients didn't have major problems with latency to begin with and didn't usually improve the latency to the CUT when OFDMA was enabled. When the airtime is highly congested, or a lot of frames are being sent to or received from many clients, the ability to win a TXOP becomes harder, and it delays the information being sent. In this setup, greatest improvements in latency were realized in high airtime usage scenarios. The greatest improvements in reduction are naturally seen when comparing to situations that have the highest latency to begin with.

In conclusion, the charts below in Figure 26 and Figure 27 summarize when an end-user may expect to see improved latency with OFDMA enabled in a house with four clients. This summary was created while generalizing the AP chipset used, ignoring differences in packet size outcomes, and generalizing results from pure Wi-Fi 6 client tests and mixed half Wi-Fi 5 client tests to provide a simpler summary with just a couple variables. The jitter or mean deviation of RTT latency and max RTT latency were heavily considered in addition to the average RTT latency to indicate if latency was seen improving or worsening. This is because the changes were often more dramatic in max RTT and jitter of RTT samples even if the average RTT didn't change meaningfully.

The color of the bars represents the CUT throughput while the four clusters of bars in each chart represent each channel utilization scenario, indicated at the bottom of each chart. The taller the bars the larger the difference in RTT latency, one way or another. If the bar, starting from the middle, is going upward, a general improvement in latency was seen. If the bar is going downward from the middle, a general degradation in latency was seen. The middle represents about the same latency observed and would be a missing bar in this set of charts, to indicate it generally was not better or worse.

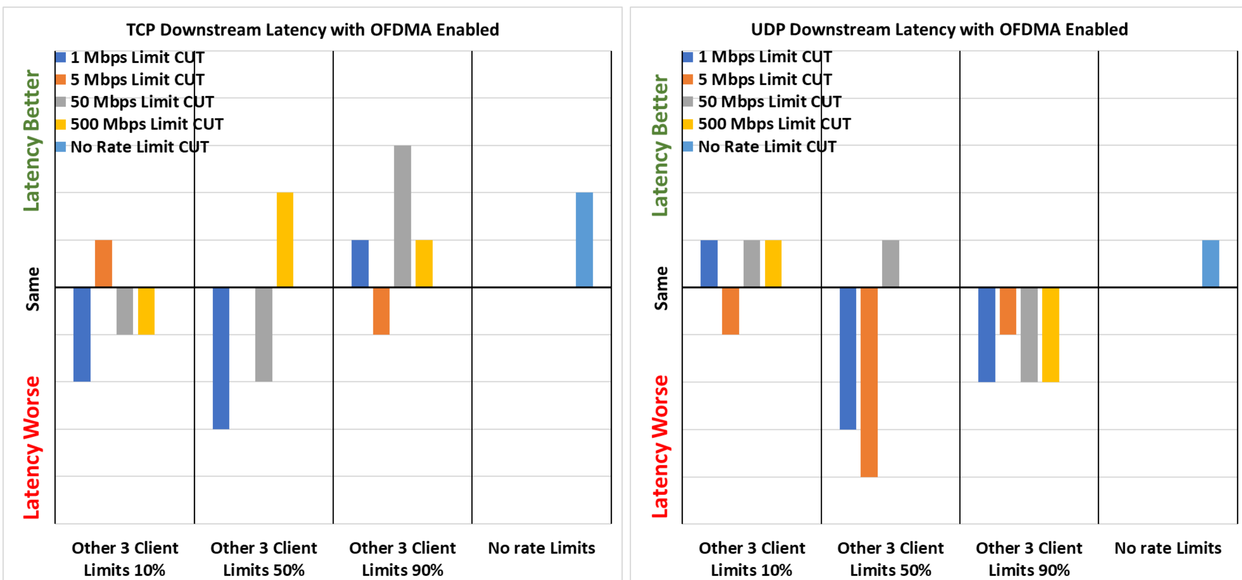

Figure 26 – Latency Change Observed with OFDMA and Downstream Traffic

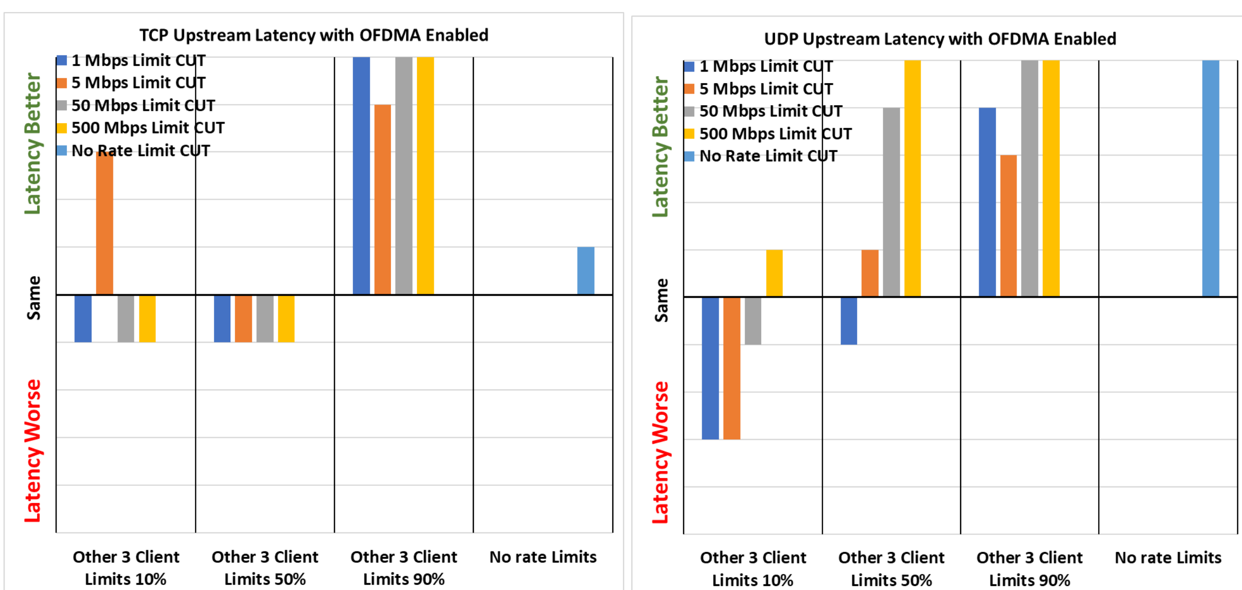

Figure 27 – Latency Change Observed with OFDMA and Upstream Traffic

The test iterations were large in number and defined many scenarios and settings, but was rigid in location of clients, number of clients, type of clients, and measurement methods. The rigid test constraints served the purpose of allowing for repeatable testing as well as allowing for relative testing so results of each mode could be compared to each other. There may be some results that are only realized in a setup like this paper assumed, and other results and improvements that were not seen from perhaps moving a client five feet in another direction, using a different client, or using 20 clients.

This paper sought to identify scenarios that improve with OFDMA enabled with the setup described herein, but it is not the only setup there is. In fact, there is not another house with the same layout, furniture, clients, AP or client positioning, or any number of many other factors which could lead to different results. There is included in this paper an extensive list of follow-up testing to consider, in order to investigate the many other ways and scenarios OFDMA can help reduce latency. In this paper, a given static setup and careful adherence to keep many things constant while iterating certain chosen parameters only, allowed for relative comparisons to be made and conclusions drawn.

Abbreviations

6E	802.11ax Wi-Fi 6 in 6 GHz band
AC	802.11ac Wi-Fi 5
ACK	acknowledgement
AP	Wi-Fi access point
AX	802.11ax Wi-Fi 6
BSR	buffer status report
BSRP	buffer status report poll
CS:GO	Counter Strike: Global Offensive by Valve/Hidden Path Entertainment
CTS	clear-to-send
CSMA/CA	carrier sense multiple access with collision avoidance
CUT	client under test
DL	downlink (AP to client) direction
DL-PPDU	downlink PLCP protocol data unit
DOCSIS	Data Over Cable Service Interface Specifications
DS	downstream (AP to client) direction
DSL	digital subscriber line
FPS	first person shooter
HE	high-efficiency 802.11ax
JSON	javascript object notation
LAN	local area network
Mbps	millions of bits per second
MCS	modulation coding scheme
mdev	median deviation
MHz	millions of hertz
MMO	massive multiplayer online
MSO	multiple systems operator
MSS	max segment size tcp data payload
MU	multi-user
MU-MIMO	multi-user multiple input multiple output
MU-RTS	multi-user request-to-send
OFDM	orthogonal frequency-division multiplexing
OFDMA	orthogonal frequency-division multiple access
OWD	one-way delay
PLCP	physical layer convergence procedure
PPDU	PLCP protocol data unit
QoE	quality of experience
QoS	quality of service
RTS	real-time strategy
TCP	transmission control protocol
TXOP	transmit opportunity
UDP	user datagram protocol
UL	uplink (client to AP) direction
UL-PPDU	uplink PLCP protocol data unit
VHT	very high throughput 802.11ac
WAN	wide area network

Bibliography & References

[1] https://www.bitag.org/documents/BITAG_latency_explained.pdf

[2] ISBN: 978-1-119-80787-2: Wi-Fi 6 & 6E For Dummies®, Extreme Networks Special Edition; David Coleman

[3] https://hpbn.co/primer-on-latency-and-bandwidth/

[4] https://www.cox.com/residential/internet/guides/gaming-performance/ping-testing.html

[5] https://parsec.app/blog/description-of-parsec-technology-b2738dcc3842

[6] https://docs.microsoft.com/en-us/skypeforbusiness/optimizing-your-network/media-quality-and-network-connectivity-performance