Wireline And Wireless Latency Improvements With AQM For Superior Quality Of Experience

A Technical Paper prepared for SCTE by

Gateek Fating
Manager, Wireless Product Connectivity
Charter Communications
6360 S Fiddler's Green Circle, CO - 80016
(818) 641-2827
Gateek.Fating@charter.com

Table of Contents

Title **Page Number**

List of Figures

Title **Page Number**

List of Tables

Title **Page Number**

1. Introduction

Internet Engineering task Force (IETF) L4S and Low Latency DOCSIS (LLD) specifications enable cable ISPs to offer low latency and low jitter services over current DOCSIS 3.1 deployments. However, successful deployment of those services depends on the effective management of latency and jitter factors from source to destination, including WiFi and access and core networks. Although many tests have been conducted to evaluate LLD with IETF non-queue building per hop behavior (NQB-PHB) and L4S traffic across the access network, the performance evaluation on real production networks is very limited. This paper will demonstrate the benefits of access layer improvement in the form of Active Queue Management (AQM) for latency-sensitive applications in queue-building scenarios. It will also assess the advantages of AQM in a production environment with a mix of queue-building and non-queue-building traffic types to assess the quantitative and qualitative gaming performance of wired and wireless private client connectivity in the presence of passive features, which increase the end customers' network throughput. The production environment will demonstrate and characterize the network gaming experience over air interface on a private client in the presence of other mobile clients with this enabled speed increase, out-of-home WiFi connectivity, etc. Not only will this paper demonstrate performance gain for latency-sensitive traffic when implementing AQM, but it will also indicate the lack of performance degradation for latency-insensitive applications that are running simultaneously.

To determine the effectiveness of AQM in improving the client experience in the production environment, network monitoring tools were used to observe baseline latency and latency under incremental load in a real-world test-house serviced by a Cable Modem Termination System (CMTS) without AQM versus a CMTS with AQM. Different qualitative and quantitative network metrics – like network latency (RTT), jitter, user input latency (application responsiveness/lag) based on frame-per-second (fps) degradation – were calculated for a cloud gaming client in the presence of various congestion scenarios.

AQM provides a superior Quality of Experience (QoE) for queue-building traffic when multiple applications simultaneously contend for airtime on a user's network. In situations where the overall network utilization is higher than the actual cable modem's provisioned speed or link rate, efficient buffering of these packets at the CMTS will avoid excessive packet drops during network congestion. In these airtime bottleneck scenarios, AQM serves to efficiently process the packets at the CMTS to provide a better quality of experience by proactively dropping just enough packets to avoid queue build up from data bursts.

2. Background

Household network traffic is increasing, and it is helpful to understand that connection speed is not the only important factor in performance. To preserve the customer experience, end-to-end latency optimization has become increasingly critical.

The delay in network communications is referred to as **network latency**; it indicates the amount of time it takes for a data packet to traverse across the network. High latency networks have a longer delay or lag, whereas low latency networks offer quick reaction times. When cloud-based applications are used for performing basic day-to-day operations within a household or business, the lag time/delay in the network response can cause deficiencies in the system. Higher latency applications degrade the overall user experience and so, although all network devices favor low latency, it is crucial for certain streaming applications like online/cloud gaming, AR/VR, online betting, real-time auctions and video-enabled remote operations.

To gauge the overall network experience from the end user's perspective, customers and industry reports also use **jitter, packet-loss and throughput** as crucial factors in determining the network performance.

- **Jitter** is the difference in time between data transmission and reception from source to destination. It has a greater influence on customer experience since it is the difference between the minimum and maximum delay. For a better user experience, a consistent delay is favored over delay changes.
- **Packet loss** is the measurement of packets that fail to reach their destination. End users may experience loss of network connectivity or slow service. Inadequate signal strength, network congestion or excessive system noise causes packet loss during data transmission. Real-time remote processing operations, like endoscopy cameras and drones for search-and-rescue operations, suffer the most when undergoing packet-loss.
- **Throughput** is the average amount of packets that can be sent/received through a network from source to destination in a given amount of time. It represents the number of data packets that successfully reach their destination despite network interference/congestion and packet loss.

This paper demonstrates the end-to-end latency improvements due to AQM deployment on the CMTS leveraging the DOCSIS-PIE algorithm. The analysis is performed in a production test house to characterize the network KPIs of a cloud gaming client via wired and wireless interface when there is an incremental load/traffic congestion. We will also discuss some of the opportunities to improve the ability to deliver improved overall quality of service on access and the WiFi network.

3. Shifting the Focus to "Working Latency"

A new way to measure real-world latency — referred to as working latency — is to look at what happens to latency when the link is utilized heavily by other clients on the network. However, the link's capacity need *not* be fully leveraged to see potential dramatic increases in working latency, as this can happen under normal usage conditions as well.

Network congestion is *not* vividly perceived by the end user unless there is an unexpected performance issue while using a latency-sensitive application that consumes significant bandwidth. Application providers use Adaptive Bitrate Streaming (ABS) by adjusting the streaming quality to detect the end client's CPU capacity and available bandwidth. End users often blame it on the network capacity in these conditions, but it is most likely the increased network latency resulting in a poor quality of streaming experience due to various factors in packet delay from the client all the way to the core network.

3.1. Types of Latency (RTT)

Idle Latency

Idle latency is the time it takes data to get from its source to destination *without any network congestion*. Idle latency measurements are beneficial to measure packet transmissions over a network path based on distance alone, but it doesn't help understand the latency under load.

Working Latency

Working latency is the time it takes data to get from its source to destination *when the network is congested.* It is a real-world measure of responsiveness when the home network connection is actively used and burdened with congestion from different clients on the network.

Working latency first determines if the network is active and then measures the time it takes for packets to reach the destination. It is a quantitative characterization of the delay for latency sensitive traffic, like video

conferencing or cloud gaming, due to the congestion/induced interference from “greedy” applications, like video streaming and file downloads.

In a nutshell, working latency assesses how strongly other internet traffic can interfere with your video conference or gaming experience. Most computer networks are inactive for most of the time. When measuring latency on an idle network, it is clear that the best-case latency is going to be delivered. This is analogous to using Google Maps travel times outside of peak hours and presuming that it represents the traffic conditions at other times of the day, which does *not* determine the network’s ability to handle congestion or reliably process the buffer bloat issue.

Network Congestion and Buffer Bloat

When a network is overburdened with more data packets than it can handle, it is said to be congested. When too many communication and data requests are made at the same time across a network that does not have the network bandwidth to transmit them, data traffic backup develops.

While **network congestion** is normally transient, it can generate annoying network issues that hinder performance, such as high levels of jitter, packet loss and delay, as well as a drop in throughput. A crowded network might indicate a broader problem in your network. As a result, it's critical to have network performance monitoring technologies in place that can detect network congestion both within and outside of your network.

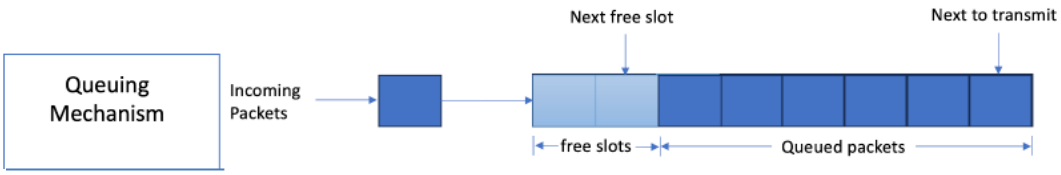


Figure 1 - General Queuing Mechanism

Buffer Bloat: Most modern network equipment, such as routers and switches, have several queues, each with a buffer to hold incoming packets. When high speed (classic TCP-based) applications put an excessive strain on the network, the queues begin to fill up and block real-time data. As a result, there is a significant delay, and perhaps fluctuating packet delays, also known as latency and jitter. This significantly degrades the experience of these interactive and latency-sensitive applications, if not totally breaking them ('buffer bloat'). Back in the days when memory constraint was not a consideration, buffer bloat was not a problem. Big in-transit buffers were designed to handle network congestion and that was not the most efficient way to tackle the buffer bloating issue. Increasing the buffer-size results in an increase in queuing delay and so Internet Service Providers (ISPs) have transitioned to an efficient way for queue management of incoming packets downstream to the CMTS.

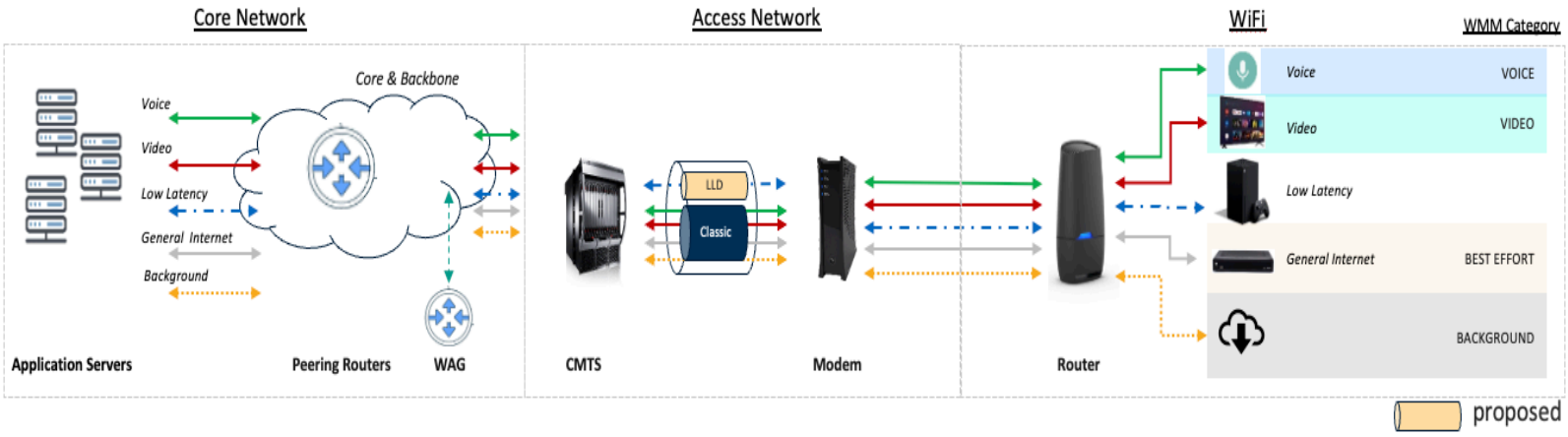


Figure 2 - Various links on network introducing latency at different stages

Figure 2 above shows the three segments within the network: **core network**, **access network** and the **in-home WiFi network**, where latency is introduced due to some of the major contributing factors like the transmission medium, propagation delay and queuing delay aspects in the network. As part of Low Latency, Low Loss and Scalable Throughput (L4S), a proposed LLD service flow would help isolate the low latency and high capacity traffic during peak network congestion.

Transmission Medium: As the packets move across the transmission medium, they introduce latency depending on the type of medium. Fiber and copper have different latency properties due to physics and so each time the network shifts from one media to another, a few milliseconds are added to the overall transmission time.

Figure 3 - Coaxial and Fiber Transmission mediums

Propagation Delay: All transmission components contribute to the propagation delay between the source and destination. All transmission components in the link affect the propagation delay of the packet. The amount of distance traveled in the medium results in propagation delay. Some types of Distributed Access Architecture (DAA) like Remote PHY (R-PHY), target to remove propagation delay by shifting/migrating the physical layer closer to the edge of the access network.

Figure 4 - Propagation Delay

Queuing Delay: This solely depends on the queuing mechanism at the cable modem as well as the CMTS. Packet congestion due to various traffic types that fill buffers within the service flow results in queuing delay. Section 4 of this paper explains queuing mechanism in detail.

Media Access Delay: In saturated network conditions, the probability of collisions and cross transmission increases resulting in high packet loss between two nodes. Asynchronization in data transmission on multiple frequency channels simultaneously, results in Media Access Control Delay.

4. Latency Improvements by Active Queue Management (AQM)

As more ISPs consider the idea of buffer bloat, interest in remedies to the problem grows. AQM is currently the most promising technology since it has the potential to produce significant network wide gains by focusing on a small number of bottleneck network components on the access network (e.g., cable modems and CMTS). The first key step in reducing latency and ensuring consistency from the CMTS to the cable modem can be achieved by implementing AQM and setting the right latency target.

Current AQM algorithms determine an estimated queueing delay of the CMTS queue and thereby calculate a packet drop probability. This drop probability is applied to incoming packets, followed by a recalculation of the drop probability. If a TCP packet is dropped, TCP congestion control takes affect and limits the throughput.

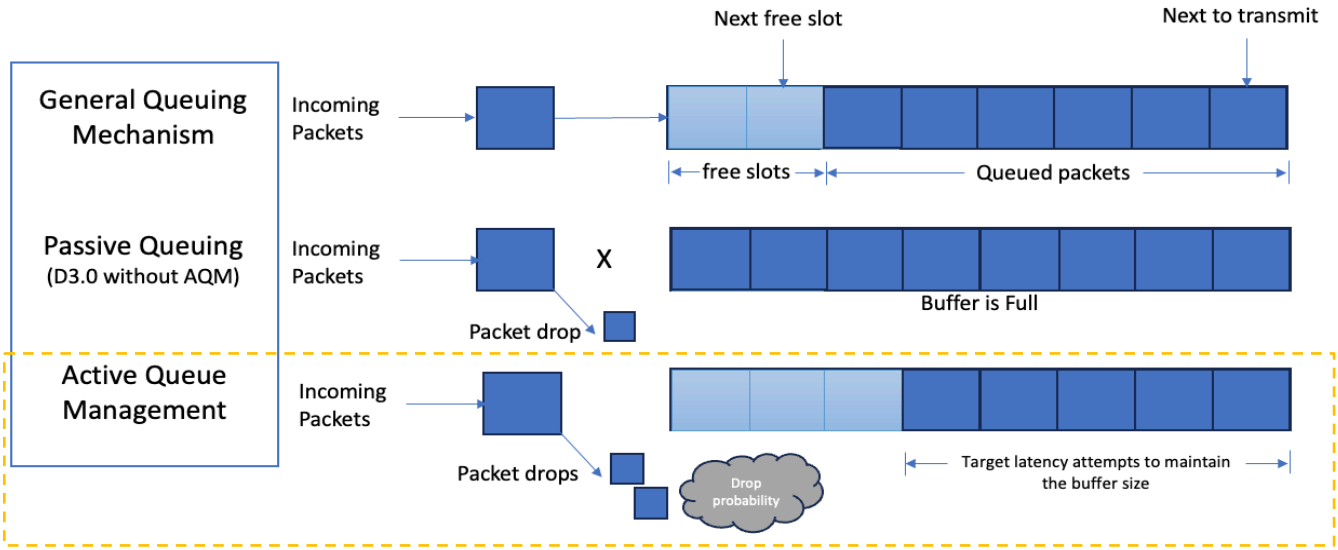


Figure 5 - Utilizing AQM for Buffer Size Management

Although an AQM technique that loses packets may be regarded unfavorable, it does reduce buffer bloat while having no major impact on any active applications. For example, a TCP session will slow down when a packet is lost, which reduces TCP traffic speed slightly, although this is typically not as essential as having high latencies and jitter for latency-sensitive real-time applications like video conferencing. When the technique employs marking instead of dropping, it is even better because it requests the sender to slow down without causing packet loss.

As part of this paper, implementation of AQM in a real-world environment at the test house is seen to radically improve the overall broadband user experience for an interactive application like cloud gaming in the presence of a congested environment. The need for AQM arises because of the presence of packet buffering in network elements and due to the mechanics of the TCP congestion avoidance algorithm.

4.1. Overview of Different Queue Management algorithms

4.1.1. Passive Queue Management

Tail drop is a queue management algorithm that is an example of passive queue management. As per this algorithm, each packet is assigned the same priority and there is no distinction. In addition, the queue length and the basis of managing the queue is concerning, i.e., when the buffer capacity is reached, inbound packets are discarded/dropped indefinitely without any delineation between latency sensitive and latency tolerant applications. This is what the algorithm looks like:

```
while packets arrive
if (queue is not full)
then
    Enqueue the packet
else
    Drop the packet
End
```

4.1.2. Active Queue Management

In this type of queue management technique, the cable modem and CMTS will keep a close eye on the incoming packets, as well as the buffer size, to keep the queuing latency under check. As soon as CMTS detects that queue-building traffic has exceeded the target queuing latency, AQM will drop just enough packets randomly to maintain the target latency, allowing more appropriate buffer levels to be maintained as part of an efficient queue management.

Table 1 - Comparison between WRED and PIE

WRED algorithm (Weighted Random Early Detection)	DOCSIS – PIE algorithm (Proportional Integral controller Enhanced (PIE))
Initialization: Empty (queue) **while** packetsarrive **if** (queue is not full) **then** Enqueue the packet **else if** (Minimum WRED threshold has been exceeded AND queue is **not** full) Execute packetdrop(newpacket) Calculate new drop probability **else** Drop newpacket **End**	Initialization: Empty (queue) **while** packetsarrive **if** (queue is not full) **then** Enqueue the packet **else if** (AQM latency target has been exceeded AND queue is not full) Execute packetdrop(newpacket) Calculate new drop probability **else** Drop newpacket **End**

Two well-known Active queue management algorithms are Weighted Random Early Detection (WRED) and DOCSIS Proportional Integral controller Enhanced (PIE). The major difference between WRED and PIE is that WRED has a concept of "drop probability" which is a function defined between *minimum threshold* and *maximum threshold*. As the queuing latency increases, the drop probability increases as well until the queuing delay reaches maximum threshold. Whereas DOCSIS-PIE defines threshold to maintain **target latency**, which in-turn sets the buffer queue size and drives the overall packet processing/queuing.

The two AQM algorithms above look similar, but "packet drop" as a function has a chance of dropping a packet, and the chance is dynamically calculated based on queue depth/buffer size. For example, if WRED, defined from 100 to 200ms, has a "packetdrop" function as per the above algorithm, it will have a 50% chance of dropping the "new packet" if the current queue depth is 150ms. In other words, we have the maximum drop rate set to 100% at 200ms, which means that 200ms is the full size of the buffer and anything beyond 200ms is tail dropped.

5. Cloud Gaming Latency Characterization Pre-AQM and Post-AQM

In this section, we will assess the network KPIs of a cloud gaming client, connected in a real-world environment without AQM and with AQM enabled on the CMTS. Different network metrics were measured as part of this analysis in a real-world environment at a test house to gauge the benefits/improvements for the latency sensitive applications under an idle environment and a congested environment with incremental load. This has also helped us understand the susceptibility of access network to congested traffic and the efficient response of the network to NOT degrade the queue building traffic due to implementation of AQM. Here are the network Key Performance Indicators (KPIs) calculated as part of this assessment.

5.1. Network KPIs For AQM Characterization

5.1.1. Monitoring the DOCSIS link continuously

This was performed using a SamKnows Whitebox tool hard-wired into the cable modem to monitor the pre-AQM and post-AQM measurements continuously. The below metrics were measured:

a) **Median Idle ICMP Latency:** This is the average Round Trip Time (RTT) Internet Control Message Protocol (ICMP) echo request of five 56 bytes packets sent from the Whitebox to the target node (on the access layer) every two hours.
b) **Median Idle UDP Latency:** This is the average RTT of a burst of User Datagram Protocol (UDP) packets transmitted to the target node every two hours.
c) **UDP Latency Under Downstream Load:** As part of this test, a 10-second downstream speed test was initiated and UDP datagrams were transmitted to the target server. Average RTT was measured for the packet delay from the Whitebox to the target server.
d) **Packet Loss (in %):** This was measured for both ICMP and UDP packets that were not received in response.

NOTE: All the above measurements were performed every two hours. Idle latency measurements were only recorded when there was no congestion detected on the network. If congestion was detected, that measurement was skipped and a retry was attempted in the next two hours.

5.1.2. Monitoring the Cloud Gaming QoE

The below metrics were calculated to assess the cloud gaming QoE, both pre-AQM and post-AQM:

a) **Average ICMP Latency/RTT:** This was measured as the average delay for gaming packets to traverse through the network and back.
b) **Average RTT Jitter:** This was the variance in arrival of two packets back from different hops on the network.
c) **User Latency OR Input to Action Latency:** The delay between the end user stimulating an input/action in the game to shoot a bullet (e.g., left mouse click) and the user experiencing the corresponding action on screen. For example, a response on the screen of a muzzle flash to the input action of a bullet shot by the end user.
d) **Cloud Gaming Throughput:** Throughput on both the downlink and uplink generated by the gaming client playing a Destiny 2 game. This is around **40Mbps on the downlink** and around **200Kbps on the uplink**.

5.2. Pre-Requisites to AQM Enablement Verification on the CMTS

Pre-requisites: When validating AQM enablement on the CMTS, we ran the following command to receive the output, as per the two screenshots below:

```
<cmts-name>#show cable service-class 2 ver | i aqm
```

```
          #show cable service-class 2 ver | i aqm
 aqm-latency 32
 aqm-enable 1
 aqm-algorithm 1
 immediate-aqm-max-threshold 1000
 immediate-aqm-range-exponent 19
```

Figure 6 - AQM Enabled on the CMTS

```
#show cable service-class 2 ver | i aqm
 aqm-latency 10
 aqm-enable 0
 aqm-algorithm 0
 immediate-aqm-max-threshold 1000
 immediate-aqm-range-exponent 19
```

Figure 7 - AQM Disabled on the CMTS

For downstream traffic, one of our Converged Cable Access Platform (CCAP) providers used the DOCSIS-PIE AQM algorithm. However, it was disabled by default, thus when users overloaded their downstream service flows with downstream traffic, latency on the service flow rose resulting in buffer bloat. Additionally, for our validation at the test house, we temporarily set the DOCSIS-PIE AQM with a **latency target of 32ms** and **maximum buffer depth to 160ms**, after which the algorithm was designed to increase the drop probability to signal the sender to reduce their sending rate or bytes-in-flight, thereby reducing overall queuing delay on the access network at higher network congestion.

5.3. Testing Methodology and Use Cases

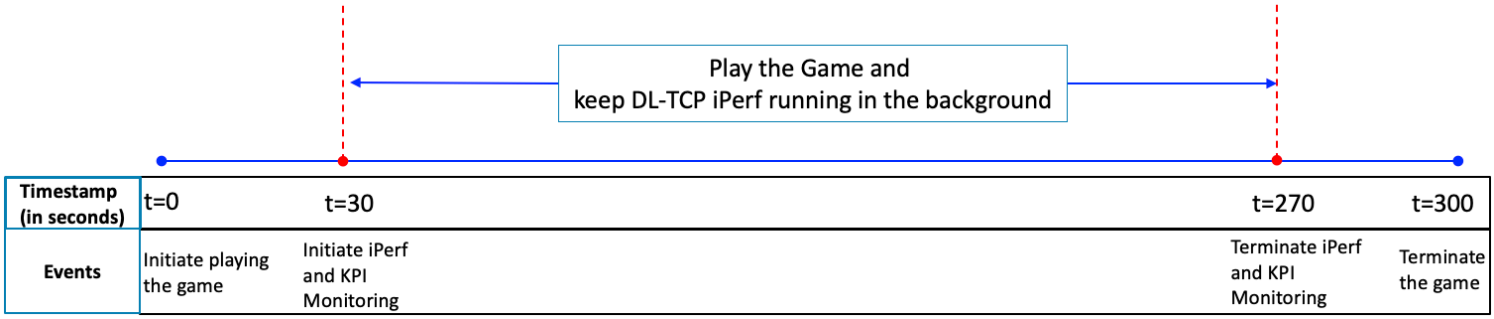


Figure 8 - Testing Methodology to Evaluate the Cloud Gaming Quality of Experience

The overall cloud gaming QoE was assessed by initiating the game and measuring the baseline latency. Following baseline measurement, incremental traffic from the other two mobile clients was induced to create contention for airtime for the latency sensitive cloud gaming traffic.

For each scenario, three iterations of four minutes (240 secs) each were performed to average the cloud gaming network performance without AQM and with AQM enabled on the CMTS. The gaming session was initiated 30 seconds before the actual measurement to ensure gaming traffic was prevalent on the link and healthy prior to introducing network congestion from the other mobile clients.

Gaming characterization was performed with the gaming PC connected via wired interface for some scenarios and wirelessly in near field RF conditions (without any lab constraints). To elaborate, when the gaming PC was connected wirelessly in near RF conditions, we maintained the RSSI signal levels between -33dBm to -36dBm and the interference between 2-3% during the entire test duration.

5.4. Network Topology for Private Gaming Characterization

The network topology below is the setup used for validation of the benefits of AQM enablement on the network. The results illustrate the improvements in section 5.5.

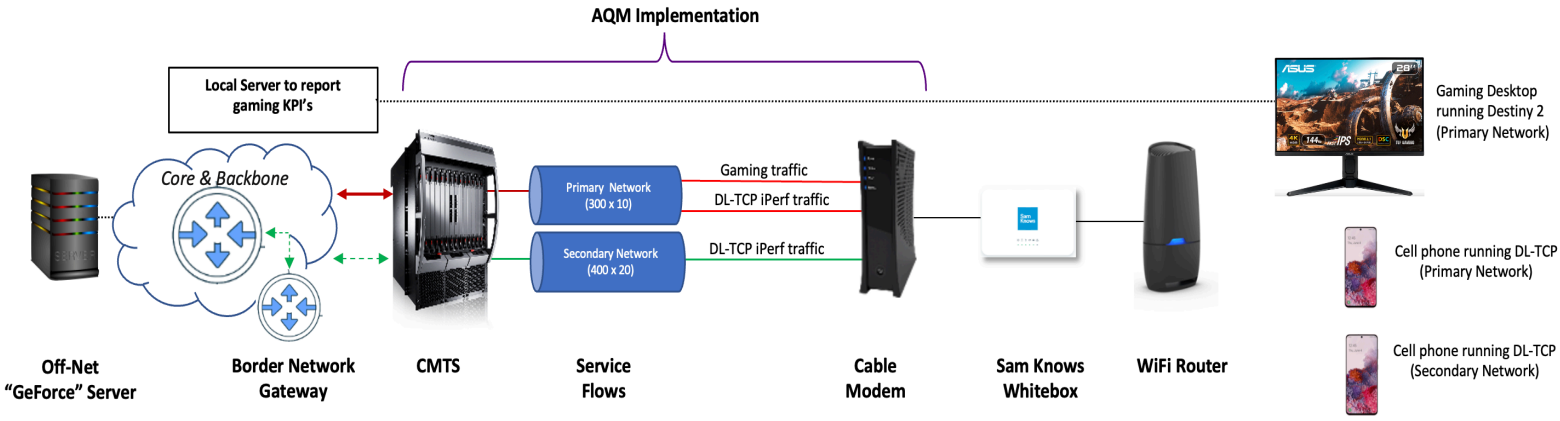


Figure 9 - Network Topology to assess the Cloud gaming latency due to downstream congestion

As per the above network topology in Figure 9, we used three clients in an environment that replicates a single-family residential household at our test house. Of these three clients, two clients (gaming PC and the cellphone 1) were always connected to the primary network provisioned at 300Mbps on the Downlink and 10 Mbps on the uplink, whereas the other phone (cellphone 2) was connected to the secondary network provisioned at 400 Mbps on the downlink and 20 Mbps on the uplink. Network KPI measurements were performed on the gaming PC to assess the impact of the cloud game with incremental throughput from two phones connected to the iPerf server.

The SamKnows Whitebox monitored the access layer for any changes to the link in case of incremental load on the private network. The tests were scheduled in the background every two hours and configured in a way to not affect the end client experience if they were already "busy." The gaming PC and the cell phone 1 were connected on the private network aka primary SSID, traversed through the "classic service flow" (i.e., the cable modem provisioned flow of 300 x 10) whereas the secondary network traffic injected by the cellphone 2 took the separate service flow (provisioned on the network for 400 x 20).

The cloud game used for this assessment was "Destiny 2" played on Nvidia GeForce Now. Additionally, the gaming KPIs were stored in our local server whereas all the SamKnows network KPIs were measured at the test target nodes located at major peering locations around the world. These peering locations are sometimes on-net or off-net, whereas the results recorded by the white boxes are stored at the data collection servers managed by SamKnows.

Figure 10 - Network Monitoring for KPI's During the Game

Latency monitoring for the gaming packets sent over the network with mean, max and last latency measurement for comparison during the game

Frame rate and fps fluctuation during the cloud game due to the airtime bottleneck at higher congestion

5.5. Results

5.5.1. Network Monitoring Stats Pre-AQM and Post-AQM

Median Idle ICMP Latency

For measuring the idle ICMP latency, we split this metric down to target nodes. The reason for this is SamKnows reaches the nearest peering location for an ICMP measurement, and we only filtered it down the target node on our network closest to the test location for a better understanding of the idle latency without any congestion. The below table indicates a 21% median idle ICMP latency improvement pre-AQM and post-AQM ICMP latency.

Table 2 - Median Idle ICMP Latency Pre-AQM vs Post-AQM

Pre-AQM (7/25 to 8/1)	Post-AQM (7/11 to 7/25)
24.85	19.65

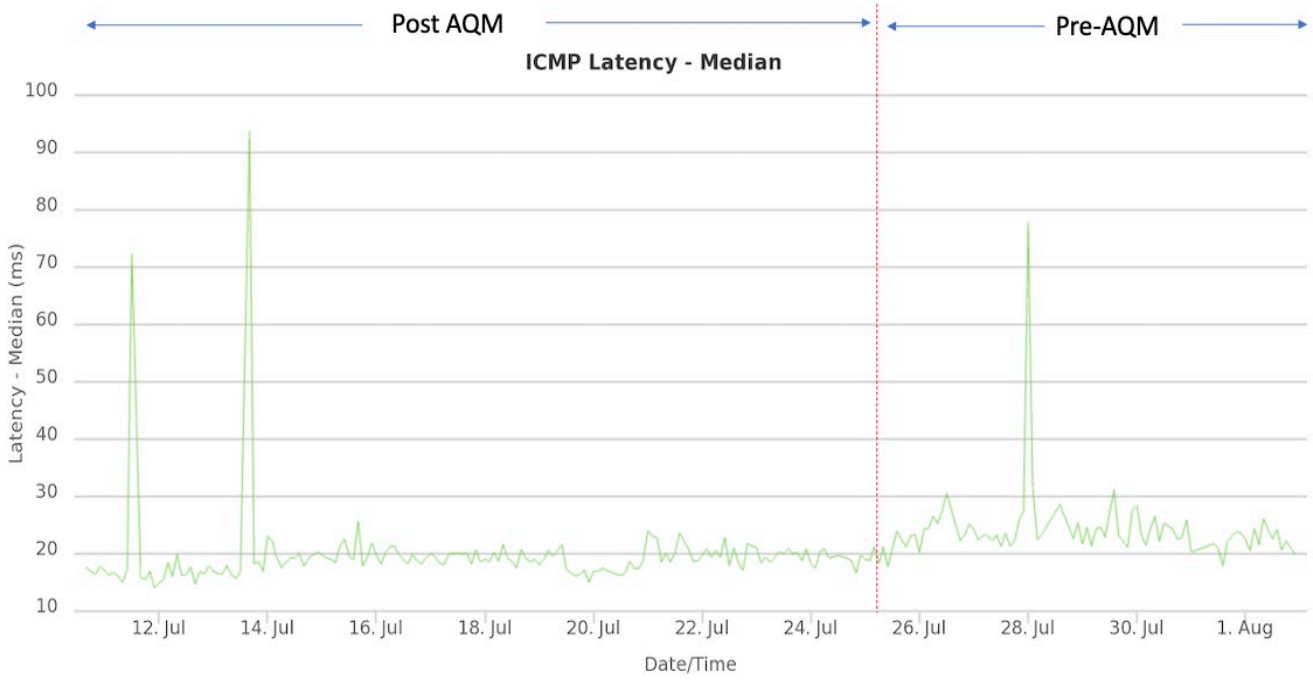


Figure 11 - Idle ICMP Latency both Pre-AQM and Post-AQM

Median Idle UDP Latency

The CDF plot of idle UDP latency in Figure 12 below indicates a clear improvement without much variance post-AQM deployment. Due to better processing ability post-AQM, idle UDP latency varies between 18.5ms to 19.5ms post-AQM deployment. Whereas UDP latency prior to AQM shows that there is a 95% probability for latency to stay between 19.7 to 25ms, but due to lack of queue management at the CMTS,

the latency numbers could also increase as high as 46ms without any load on the network. The standard deviation of UDP idle latency pre-AQM was found to be 1.96ms.

NOTE: Even though the measurement calls out Idle UDP latency, the measurement was performed by sending a burst of UDP datagrams from the whitebox to the target server.

On the other hand, post-AQM latency is seen to be remarkably improved. The idle UDP latency was confined from 18.5ms to 19.52ms (i.e., 1ms of standard variance) when the SamKnows Whitebox measured the latency over 14 days. The standard deviation of UDP idle latency post-AQM was found to be 0.18ms.

Table 3 - UDP Latency performance Pre-AQM and Post-AQM

	Range of Idle UDP Latency (in ms)	Standard Deviation (in ms)
Pre-AQM	19.7ms to 46ms	1.96ms
Post-AQM	18.48 to 19.5ms	0.18ms

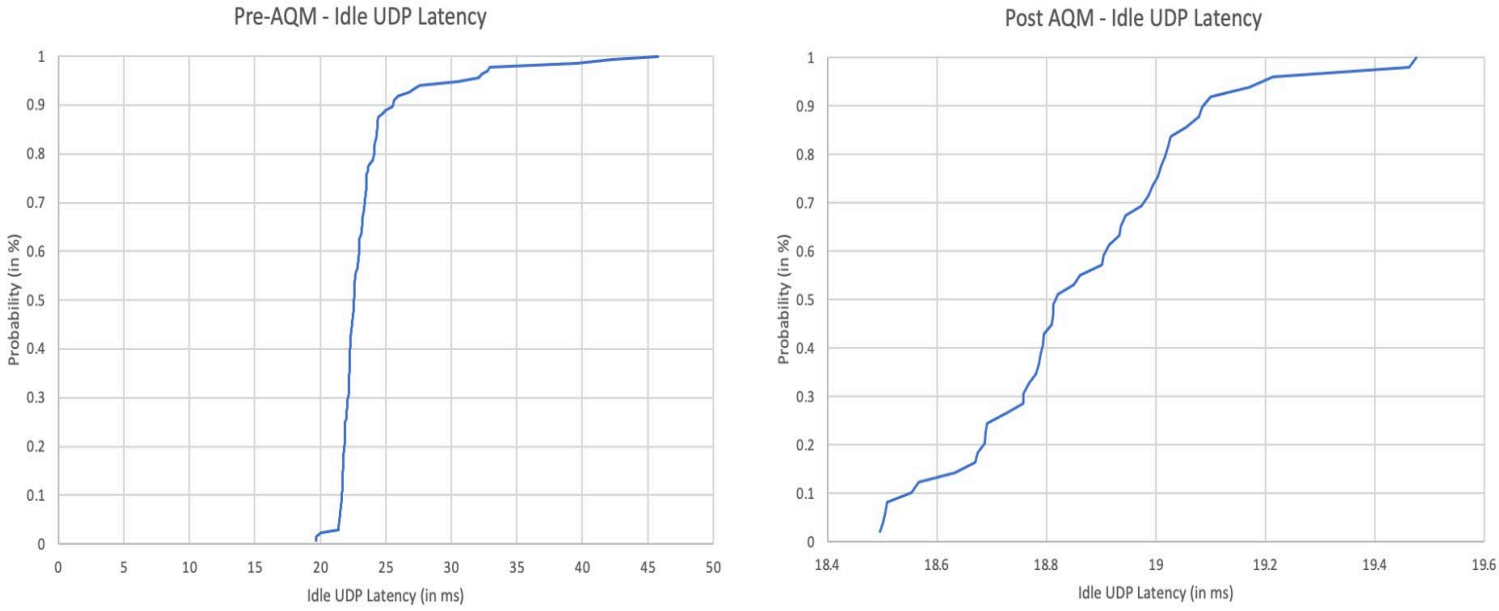


Figure 12 - CDF (Cumulative distribution function) plot of Idle UDP latency

In the below UDP latency graph, we see high jitter (variance in latency numbers in presence of queue-building traffic) prior to AQM enablement.

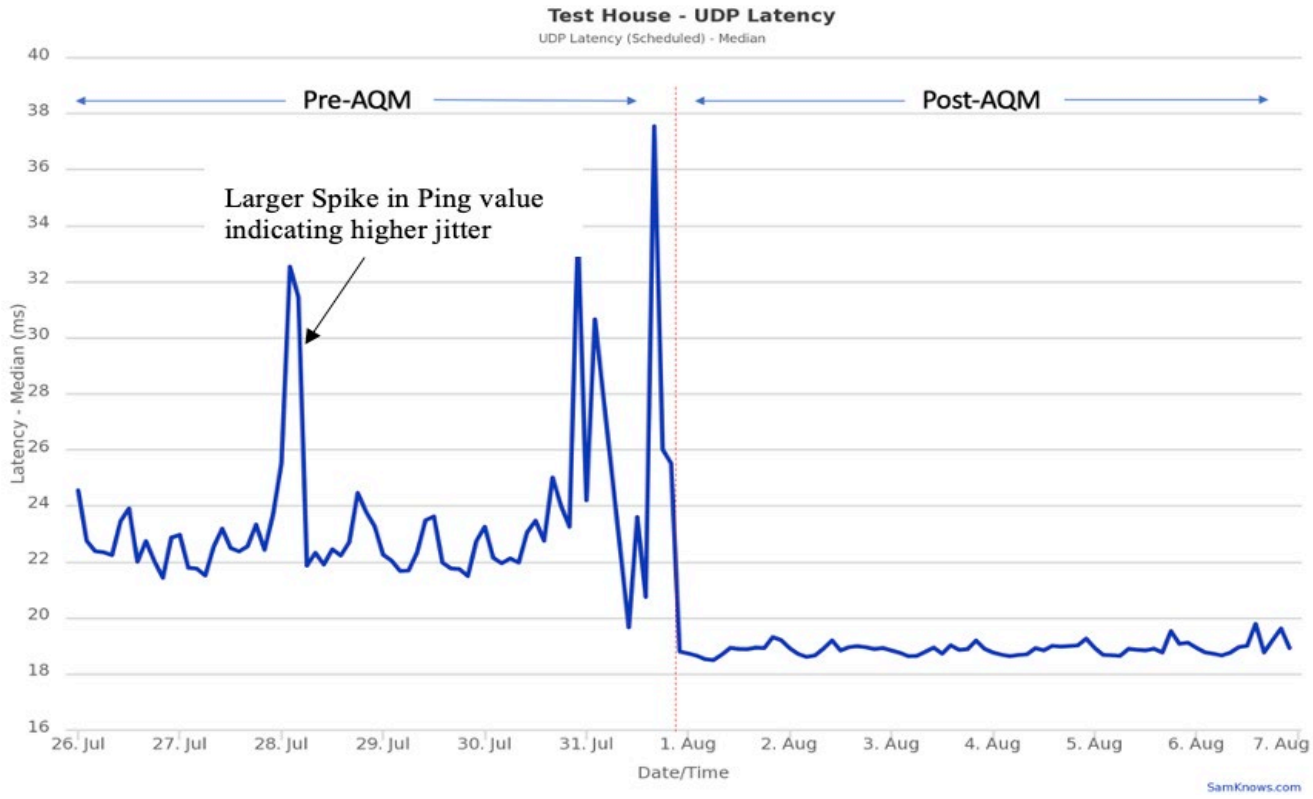


Figure 13 - UDP latency pre-AQM and post-AQM

Latency Under Downstream Load

Under a congested network, we see a clear improvement in latency by around 56% with AQM deployment on the network. Although, we see an improvement in idle conditions without much congestion. It is evident in Figure 14, below, which indicates that when TCP keeps the buffer full, AQM processes and queues the packets in the buffer as a way to slow down the TCP transmission by dropping a few packets to indicate congestion.

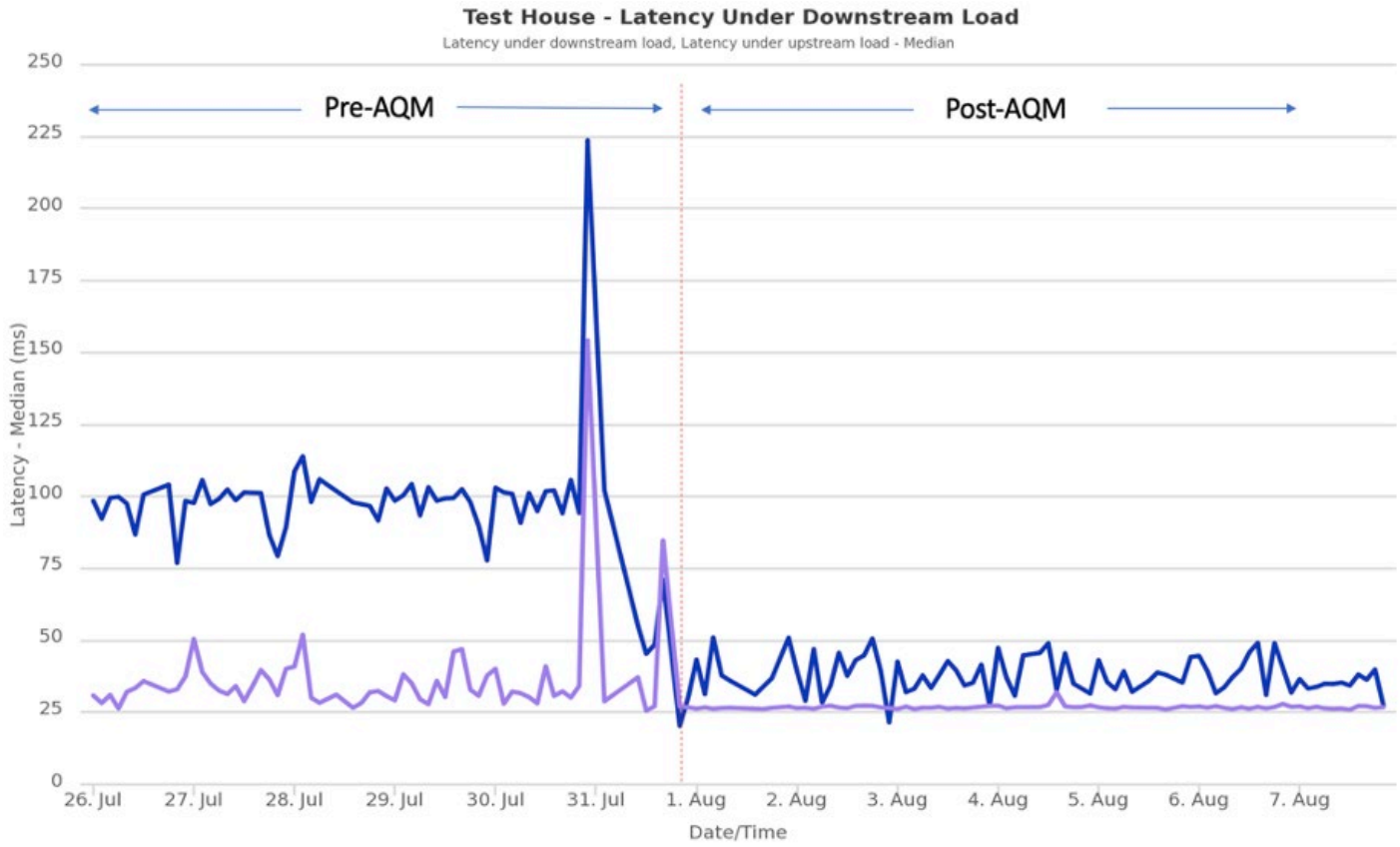


Figure 14 - ICMP Latency Under Downstream Load

ICMP Packet Loss (in %): This tracks the fraction of ICMP packets lost during the scheduled idle ICMP test every two hours to assess the health of the link.

Table 4 - Improvement in Packet Loss (%) Post-AQM

	Pre-AQM Aggregate *to 7/25 to 7/31*	Post AQM Aggregate *8/1 to 8/7*
Median	0.11%	0.02%
P95	0.36%	0.14%

5.5.2. Gaming Characterization results Pre-AQM and Post-AQM

Average Gaming Latency on Wired Interface (Pre-AQM vs Post-AQM)

With the gaming client hard-wired to the router and two mobile phones pushing incremental load on the network, after AQM enablement, gaming latency at higher network congestion is controlled due to better queue protection at the CMTS. Please note that gaming latency at higher network congestion is comparable with increased iPerf traffic from the two mobile clients.

Post-AQM Improvement (at higher congestion highlighted in the below graph) = 27%

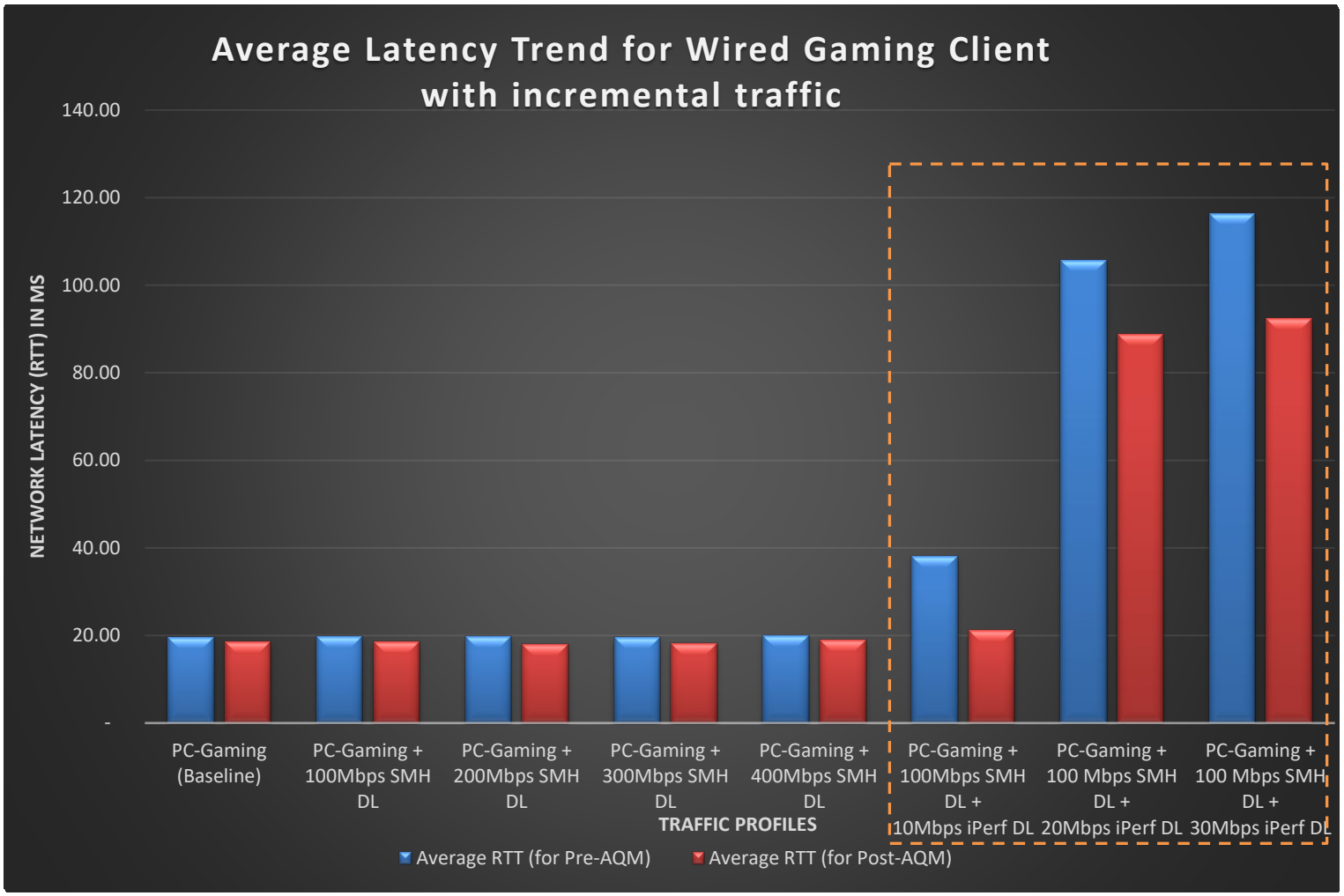


Figure 15 - Average Latency Trend for Wired Gaming Client with incremental traffic

Average Gaming Jitter on Wired-Interface (Pre-AQM vs Post-AQM)

Average Jitter stats measured during the Destiny 2 gaming session at the test house indicated a significant improvement after AQM enablement. With higher congestion on the network, we saw no negative impacts like screen rendering on the gaming experience.

Post-AQM improvement = 39.3% (at higher congestion highlighted in the below chart)

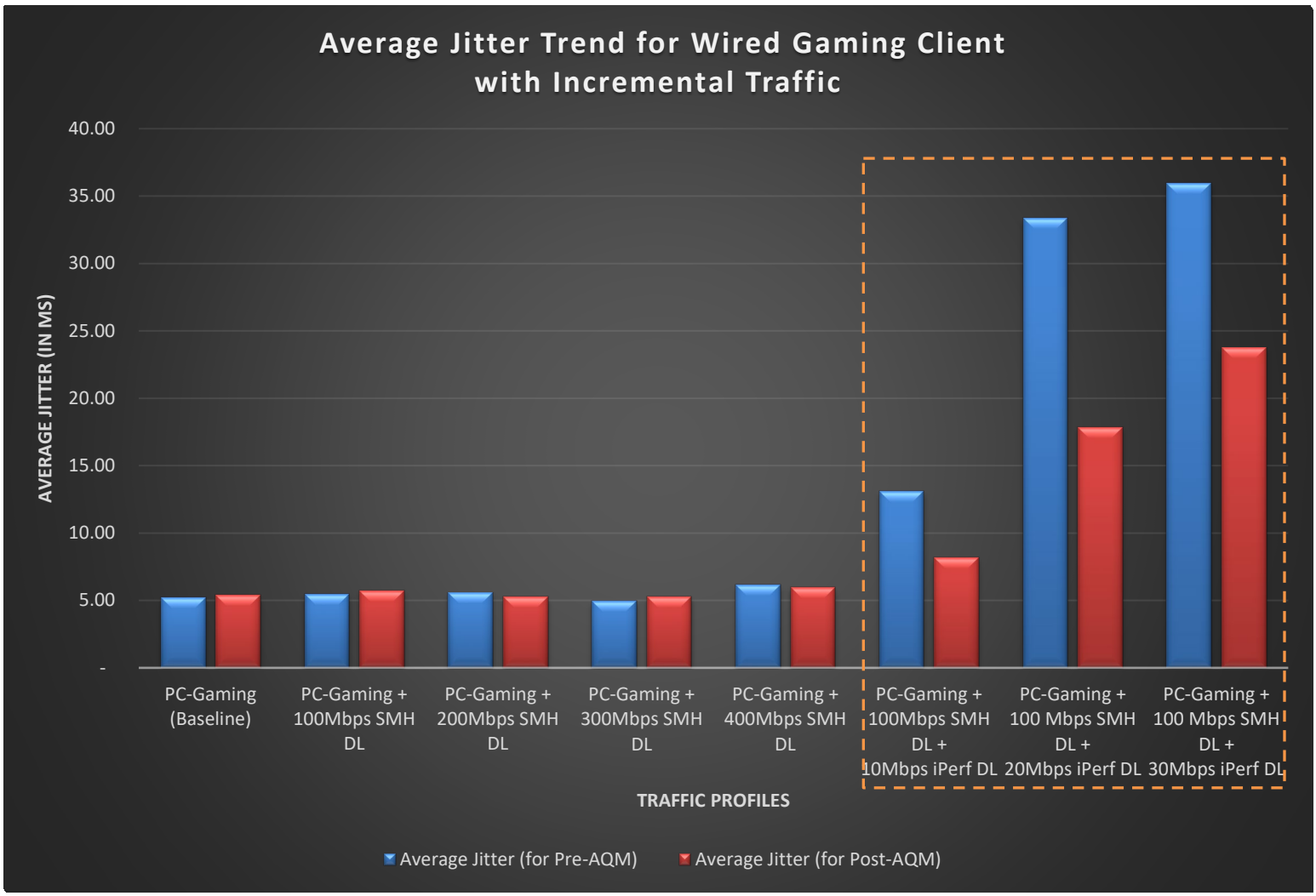


Figure 16 - Average Jitter Trend for Wired Gaming Client with incremental traffic

Average Gaming Latency for Near Field OTA (Pre-AQM vs Post-AQM)

Post-AQM latency is slightly lower than pre-AQM for all the test scenarios where we introduced additional traffic for airtime contention. In this case, the overall gaming latency trend for pre-AQM and post-AQM do not experience a huge improvement after AQM enablement on the network.

Even in other scenarios without much interference on the network, we do see comparable results. Between pre-AQM and post-AQM when the gaming PC is connected wirelessly in near RF field from the router Post-AQM improvement = 12.8%

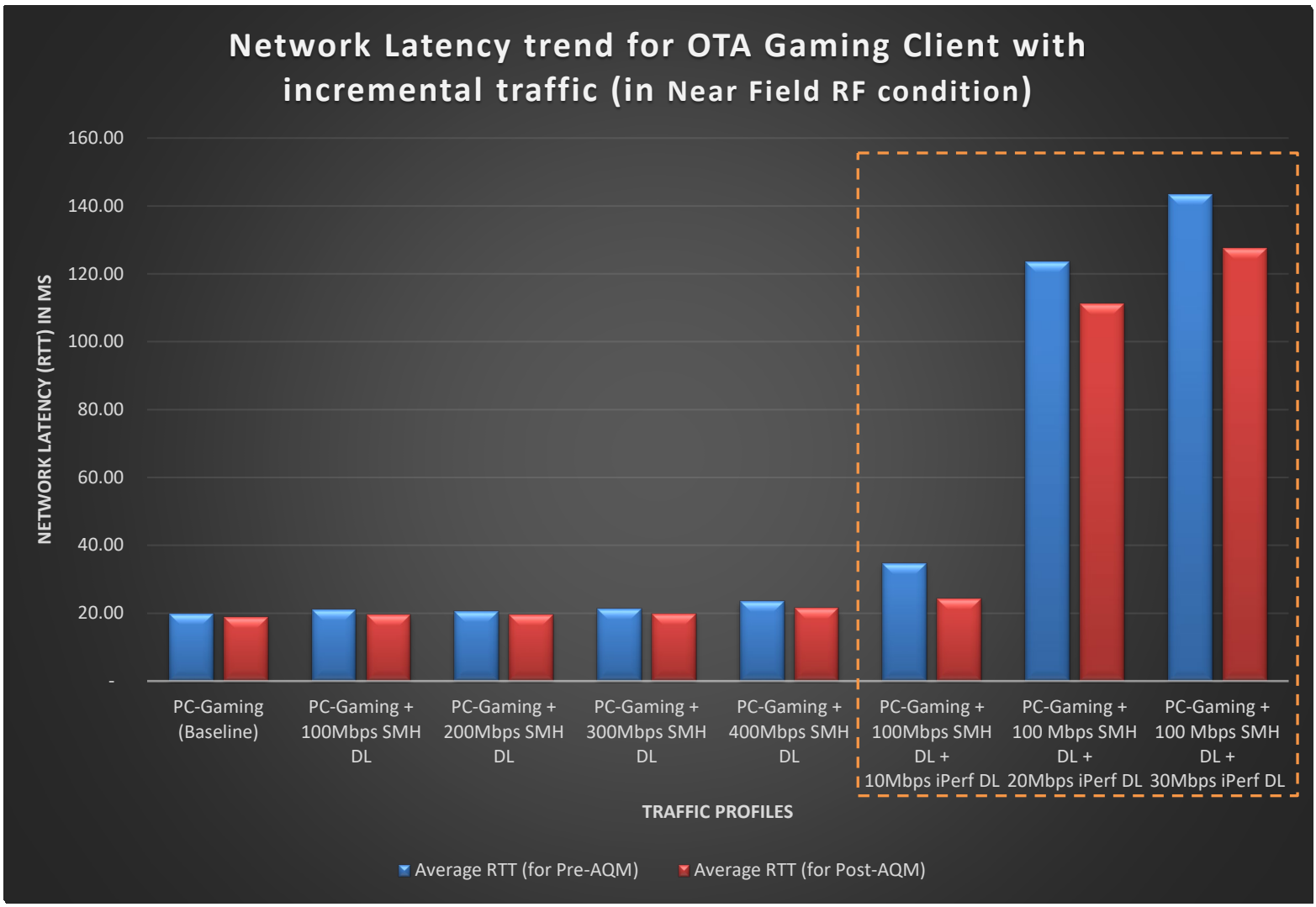


Figure 17 - Network Latency trend for OTA Gaming Client with incremental traffic (in Near Field RF condition)

Average Gaming Jitter for Near Field OTA (Pre-AQM vs Post-AQM)

Jitter is very comparable both pre-AQM and post-AQM when the gaming client is connected wirelessly in near field RF environment.
Post-AQM improvement = 7.8 %

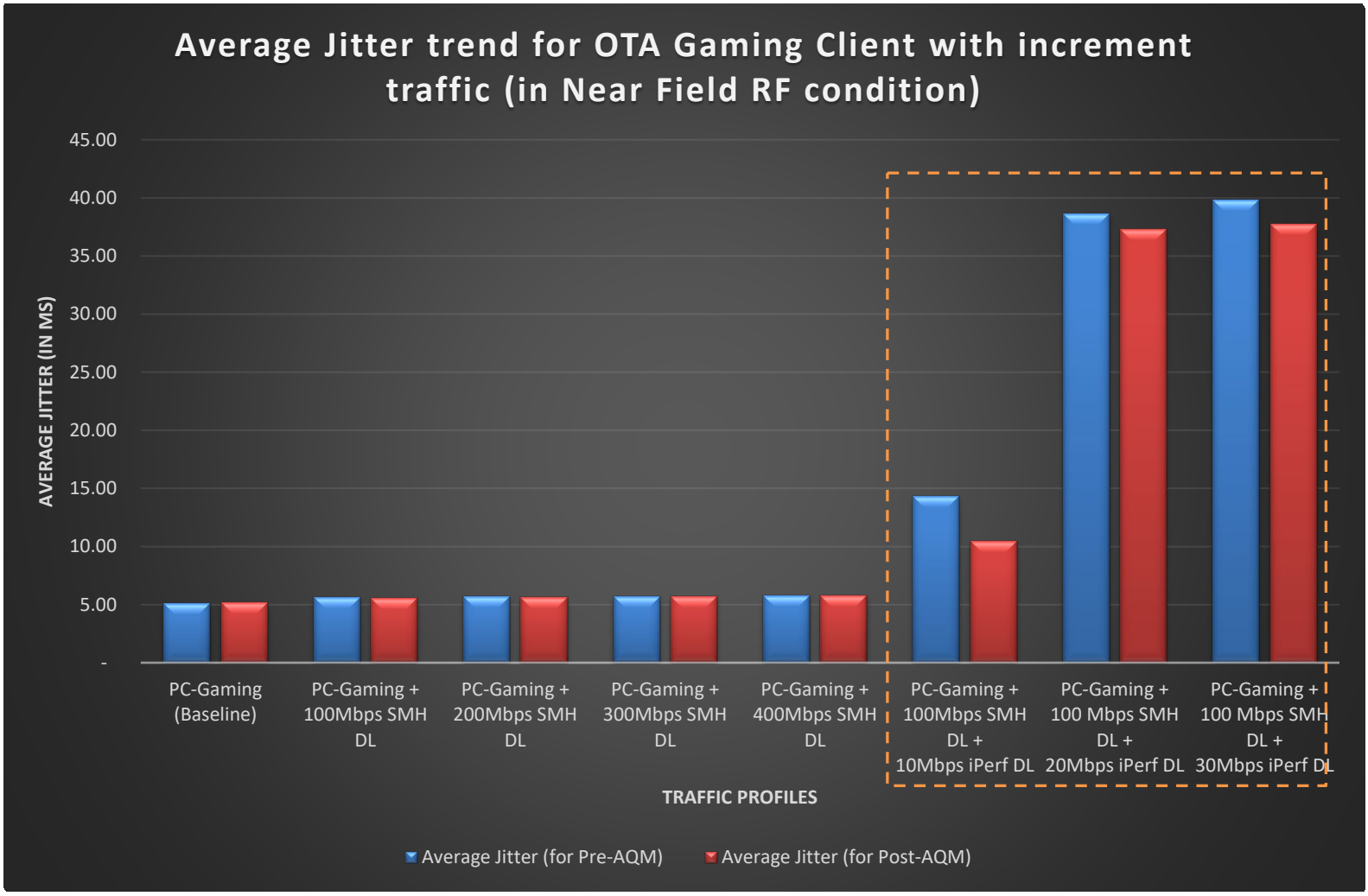
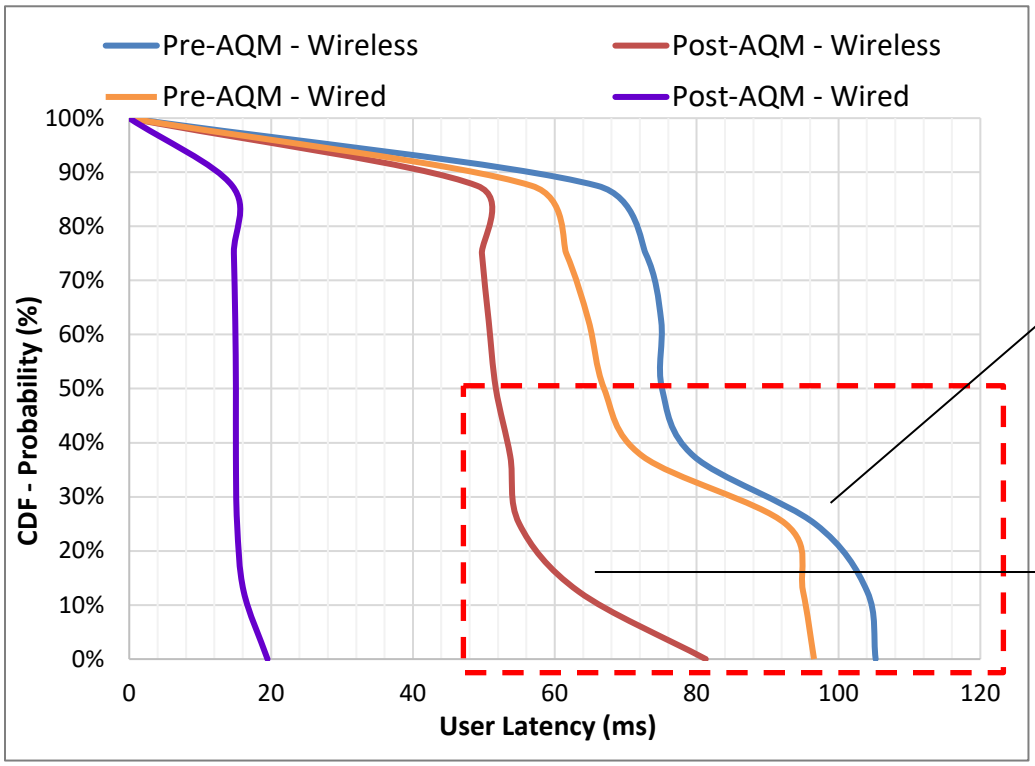


Figure 18 - Average Jitter trend for OTA Gaming Client with increment traffic (in Near Field RF condition)

User Latency: Significant improvement is seen in user latency (i.e., input to action latency). This is the reaction delay for the user to notice the fps (frame per second) change during the game. Higher user latency indicates delay in screen rendering. Also, higher congestion results in screen tearing phenomena for the end user without AQM enablement on the CMTS. The far right column in Table 5 illustrates the wired post-AQM user latency improvements by 7 to 11 times after AQM is enabled on the CMTS.

Table 5 - User Latency Comparison on Gaming Client in Pre-AQM and Post-AQM

Traffic Profile	User Latency			
	PRE-AQM		POST-AQM	
	Near OTA	WIRED	Near OTA	WIRED
PC-Gaming (Baseline)	65.99	49.05	56.73	14.51
PC-Gaming + 100Mbps SMH DL-TCP	72.80	49.72	61.61	14.75
PC-Gaming + 200Mbps SMH DL-TCP	75.01	50.68	64.77	14.95
PC-Gaming + 300Mbps SMH DL-TCP	75.12	51.69	66.94	15.04
PC-Gaming + 400Mbps SMH DL-TCP	79.61	53.72	72.38	15.04
PC-Gaming + 100Mbps SMH DL + 10Mbps iPerf DL-TCP	96.70	54.99	92.54	15.21
PC-Gaming + 100 Mbps SMH DL + 20Mbps iPerf DL-TCP	104.05	63.36	95.01	16.14
PC-Gaming + 100 Mbps SMH DL + 30Mbps iPerf DL-TCP	105.24	81.29	96.55	19.50



Convex curve indicates that at higher congestion, without AQM, queue building traffic will result in excessive packet drops and an overall worse quality of experience.

Concave curve indicates that at higher congestion, AQM monitors the buffer and adjusts its target latency to maintain the overall quality of experience

Figure 19 - Impact of User Latency with incremental throughput

Cloud Gaming throughput: With incremental congestion, as shown in the table below, we notice that the private gaming throughput is deteriorated when AQM is disabled.

Table 6 - Cloud Gaming throughput characterization (Pre-AQM vs Post AQM)

Traffic Profile	Wired		Near Field OTA	
	Pre-AQM	Post-AQM	Pre-AQM	Post-AQM
Profile 1: PC-Gaming (Baseline)	41.4	40.87	41.14	41.49
Profile 2: PC-Gaming + 100Mbps SMH	41.63	41.61	41.55	41.67
Profile 3: PC-Gaming + 200Mbps SMH DL	41.59	41.51	41.65	41.92
Profile 4: PC-Gaming + 300Mbps SMH DL	41.57	41.64	41.52	41.4
Profile 5: PC-Gaming + 400 Mbps SMH DL	41.22	41.48	41.47	41.48
Profile 6: PC-Gaming + 100 Mbps SMH DL + 10Mbps iPerf DL	26.11	41.57	26.28	41.55
Profile 7: PC-Gaming + 100 Mbps SMH DL + 20Mbps iPerf DL	26.18	41.51	26.12	41.58
Profile 8: PC-Gaming + 100 Mbps SMH DL + 30Mbps iPerf DL	26.11	41.47	26.06	41.6

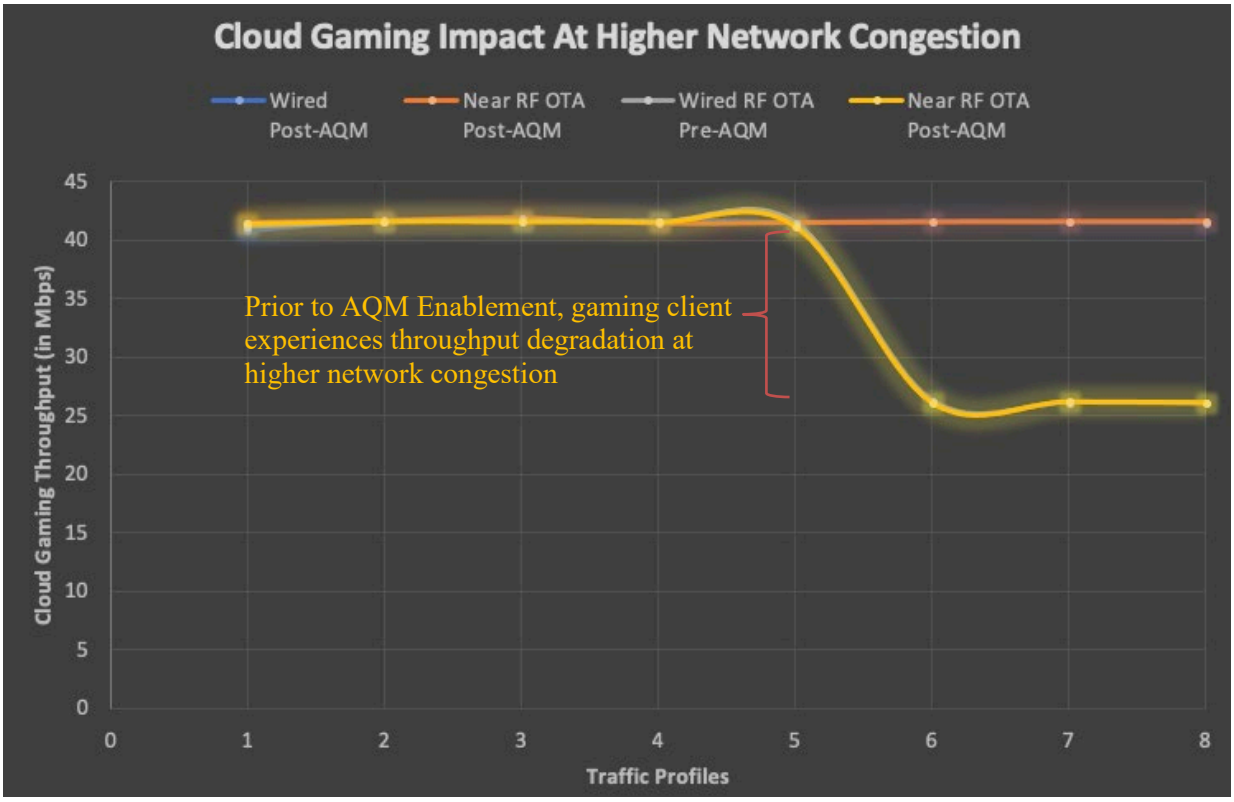


Figure 20 - Cloud gaming throughput impact at higher congestion levels

6. End-to-End Latency Management

The operational challenges of providing the best Quality of Service (QoS) solution for the user calls for an end-to-end latency improvement strategy for the client, including on WiFi so that congestion control notifications drive the prioritization and queue management of incoming packets based on the WMM access categories and implementation of L4S-AQM scheduler.

6.1. Latency Management Strategies on WiFi

With an increasing amount of mobile devices onboarding on the customer's network, WiFi routers are soon becoming a barrier/bottleneck to increasing the potential for buffer bloat on the customer's network. Although AQM implementation dramatically improves the end user's quality of experience on a wired interface, there are still opportunities for ISPs to efficiently manage working latencies on the WiFi link to deliver a superior quality of service. Latency optimization technologies exist today including Peering densification, AQM, LLD and WiFi WMM to address near to mid-term online experiences

6.1.1. Current Practice

Orthogonal Frequency Division Multiple Access (OFDMA)

OFDMA is part of the 802.11ax technology that demands very effective scheduling strategies to operate in highly dense and complicated network installations. Some of the most significant advantages of OFDMA include reduced network latencies. The entire Wi-Fi network latency in the context of WiFi networking is made up of the combination of downlink latency, or access point (AP) to client, and uplink latency, or client to AP.

Older WiFi versions allow any device in a network to commence transmission at any moment with little cooperation. While this method works well in less dense environments, it is inefficient in dense deployments owing to packet collisions. OFDMA is used in WiFi 6 to manage transmission and reception to and from non-AP devices through centralized coordination by an Access Point (AP). This enhances transmission efficiency and, as a result, lowers average latency.

WiFi Multimedia (WMM)

WMM is a WiFi specification built to prioritize voice and video traffic over best effort internet traffic while de-prioritizing background, non-latency sensitive packets on the network. In a congested environment, WMM guarantees to continue receiving the priority traffic it requires. This also ensures that latency sensitive packets spend less time in the queue, which reduces the overall latency and jitter.

According to the IEEE 802.11e wireless QoS standard, the WiFi Alliance specification allows network packets to be processed differently based on application or network tags that define the traffic type. WMM, which is based on tags sent over the network on the downstream and upstream, may be implemented efficiently in a hostile WiFi environment to prioritize latency sensitive traffic ahead and prevent congestion.

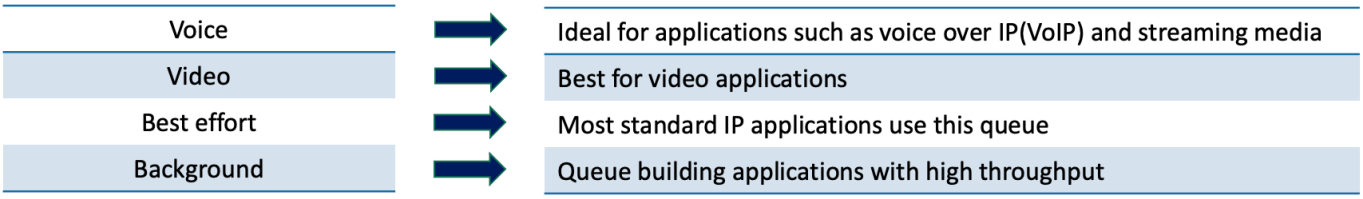


Figure 21 - Different Data-type classifications in priority order, highest to lowest

6.1.2. Future Scope

When clients are connected on the wired interface, the likelihood of latency degradation is greatly improved by implementing the right AQM algorithm. To enhance the experience further for clients on the WiFi link, other technologies like Low Latency, Low Loss, Scalable Throughput (L4S) would set the platform to achieve low latency and higher throughput for latency sensitive applications. Here are a few latency management initiatives that could greatly improve the overall quality of experience for the customer.

Low Latency, Low Loss Scalable Throughput (L4S)

The core basis of L4S is to resolve queuing delay by adopting a new/enhanced class of congestion control (assisted by a modified form of ECN) that seeks capacity with much less queuing delay. LLD's basic features include support for two queues in each direction:

a. "Classic" queue for standard congestion-controlled traffic with deep buffer (together with AQM), which enables classic congestion controllers to deliver high throughput while keeping latency under reasonable control.
b. "Low Latency" queue for traffic that does not cause delay, latency fluctuation or loss. The low latency queue contains an extremely thin buffer as well as additional features that allow "well behaved" applications to achieve ultra-low latency.

L4S extends ultra-low latency treatment to a wider range of applications and enables greater end-to-end latency management (beyond DOCSIS). L4S targets high data rate, ultra-low latency and low packet loss, which is good for cloud gaming, cloud AR/VR, etc. Implementation of L4S with AQM over WiFi should certainly be considered as a future scope on WiFi after LLD implementation on DOCSIS.

L4S uses application marking and for it to be used on WiFi, networks need to respect/carry Differentiated Services Code Point (DSCP) marking across the access and core networks. This requires industry adoption by external content providers as well as ISPs for low latency use cases.

Table 7 - Fundamentals to Support Latency Management on WiFi

Initiative	Description	Expected Action
Traffic / Application-Based Prioritization	Dynamic prioritization of traffic over WiFi based on congestion affecting user experience	Improved WiFi experience for customer dynamically based on congestion, traffic and application being used

Device-Based Prioritization	Prioritization of specific devices that are connected to the network with or without user intervention	Potential use cases could include ISPs letting customers choose the traffic prioritization on a device based on their needs
WiFi Queue Management & Advanced Traffic Prioritization	Queue management using combination of traffic, application and device-based prioritization along with leveraging WMM access categories	Improved WiFi experience for customer dynamically based on congestion, traffic, device and application being used Note: This would need to have individual implementations completed for app, device and traffic

7. Conclusion

Network latency is often thought to affect only applications that leverage higher throughput, but persistent low latency results provide a better connectivity experience for the user. Access layer latency initiatives (like AQM, LLD, etc.) improve the experience for wired client connectivity and enable the access network infrastructure to more efficiently process packets. However, to deliver an end-to-end QoS to the customer on WiFi, classic Explicit Congestion Notifications (ECN) and "scalable" congestion control to prioritize and mark the packets at the router/access point on the airlink are critical.

The validation conducted at the production test house to assess the cloud gaming proved that AQM is much more effective for latency-sensitive traffic (like cloud gaming as discussed in this paper). When queue depth or buffer size approaches the threshold target latency set by the DOCSIS-PIE algorithm, we do see an efficient approach to calculate drop probability and some packet drops to signal the sender to slow down and thereby control congestion.

Additionally, although we do see improvements on the access layer KPIs enablement on the network, the end user improvements are perceived by wired clients only. Clients connected wirelessly to the router would not vividly experience a difference in their QoE unless latency management strategies are implemented on the router/access point to mark the traffic based on its classification type. These could be done through scalable and classic congestion notifications, WMM markings and low latency service flows on WiFi.

Standardization of how the QoS metrics are measured across core, access and in-home WiFi is critical, and so end-to-end QoE assessment will help the cable industry drive across the common goal of standardized latency monitoring and optimization across the transmission medium.

Abbreviations

AP	Access Point
Mbps	Megabits per second
ms	Milliseconds
AQM	Active Queue Management
WRED	Weighted Random Early Detection
DOCSIS	Data over Cable Service Interface Specifications
PIE	Proportional Integral Controller Enhanced
L4S	Low Latency, Low Loss, Scalable Throughput
IETF	Internet Engineering Task Force
CCAP	Converged Cable Access Platform
CMTS	Cable Modem Termination System
PGS	Proactive Grant Service
LLD	Low Latency DOCSIS
DCTCP	Data Center Transport Control Protocol
WMM	WiFi Multimedia
OFDMA	Orthogonal Frequency Division Multiple Access
QoS	Quality of Service
ECN	Explicit Congestion Notification
SMH	Spectrum Mobile Home
DL-TCP	Downlink- Transport Control Protocol
fps	Frame Per Second
OTA	Over The Air
ICMP	Internet Control Message Protocol
SSID	Service Set Identifier
UDP	User Datagram Protocol
KPI	Key Performance Indicator
RTT	Round Trip Time
IEEE	Institute of Electrical and Electronics Engineers
QB	Queue Building
NQB	Non-Queue Building
PHB	Per Hop Behavior
QoE	Quality of Experience
QoS	Quality of Service

Bibliography & References

1 Improving The Latency Of An MSO Network For Gaming And Real Time Applications, by Colin Dearborn

2Low Latency DOCSIS Overview And Performance Characteristics by Greg White, Karthik Sundaresan and Bob Briscoe

3Fastest Path to Low Latency Services by Sebnem Ozer

4Configuring and Deploying Low Latency DOCSIS Networks by Greg White, Karthik Sundaresan

5Dual-Queue Coupled Active Queue Management (AQM) for Low Latency, Low Loss, and Scalable Throughput (L4S) RFC 9332 by K. De Schepper*, B. Briscoe and Greg White*

6Tracking Round Trip Time Latency in the MSO Network by Michael Overcash; Alan Skinner; Owen Parsons; Daniel Sciscoe

7Towards Predictable Low Latency DOCSIS Services by Dan Rice, Sebnem Ozer, Ph.D., James Martin, Ph.D.

8Webinar – Improving Latency with AQM June 2023 by Chris Comstock and Hamish Bala

9Webinar – Unleashing the Power of Low Latency: Enhancing WiFi Networks for Optimal User Experience by Annie George, Sebnem Ozerr and David Dickson

