SCTE·ISBE CABLE-TEC EXPO 2019 NEW ORLEANS, LA SEPT. 30-OCT. 3

2019 Fall Technical Forum SCTE·ISBE • NCTA • CABLELABS

Winning the Gaming War: Play for Cable Operator

Assuring video game experience across multiple domains

A Technical Paper prepared for SCTE•ISBE by

Alon Bernstein
Distinguished Engineer
Cisco Systems
alonb@cisco.com

Rajiv Asati
Distinguished Engineer
Cisco Systems
rajiva@cisco.com

Sangeeta Ramakrishnan
Distinguished Engineer
Cisco Systems
rsangeet@cisco.com

Table of Contents

List of Figures

List of Tables

Title **Page Number**

Introduction

There are people who play games and there are people who watch games. While it is debatable whether Video Gaming is a sport, it is a fact that Video Gaming rivals traditional media as a form of entertainment, whether offline or online (i.e. available over the internet). As Netflix stated in its shareholder report that "We compete with (and lose to) Fortnite more than HBO" (see ref [1]).

As online gaming (mobile, PC, console, etc.) continues to exponentially increase (see ref[10]), the network performance (not just availability) is critical for the superior game experience and in particular to action games such as FPS (first-person shooters) and e-sports in multi-player mode. Arguably, network performance may be more critical for cloud gaming such as Google Stadia, since the gaming experience solely relies on the cloud (for almost all of the processing) over the network. The network performance (mainly, bandwidth) is also somewhat critical while broadcasting one's game in real-time on any of popular social platforms e.g. twitch, youtube etc. for others to watch (note that online game watching is the second most popular viewing with ~100 million viewers, more popular than MLB, NBA or NHL per ref[12]).

This paper covers the challenges in getting an excellent online game experience for action games, not only from the game developer and players point of view, but also from the network point of view. The paper does not focus much on game streaming (e.g. Twitch) or Cloud Gaming (e.g. Stadia).

As we discuss in the paper, the online game experience wrt the network is multi-domain by nature – home network domain, service provider network domain, Internet domain and the data center network domain (where the game servers are located) are all important parts, each one with its own set of technical and non-technical issues. Of course, there are other challenges in ensuring superior game experience outside the network, such as the game engine, match making, rendering rate of a graphics card etc., but these are not the focus of this paper.

We see a trend of service providers marketing "low latency" as a differentiator that is essential for gameplay. They equate "Lag" to "Latency". It intuitively makes sense. Isn't multi-player gaming similar to a dual where the fastest to draw is likely to win? As we explore in this paper, the answer is more complex and depends on the type of lag compensation algorithms used by the game server, and what the gamer defines as "Lag" is not exactly what a network expert defines as "Latency".

The above has been the trailer to our paper, now let the game begin!

Overview

Online Video Gaming has attracted the attention of network engineers ever since the first multi-player online games appeared. Arguably, Online gaming has become one of the largest $$$ businesses on the Internet since 2017 and growing with the high CAGR, as illustrated in the figure below (see ref[17]) -

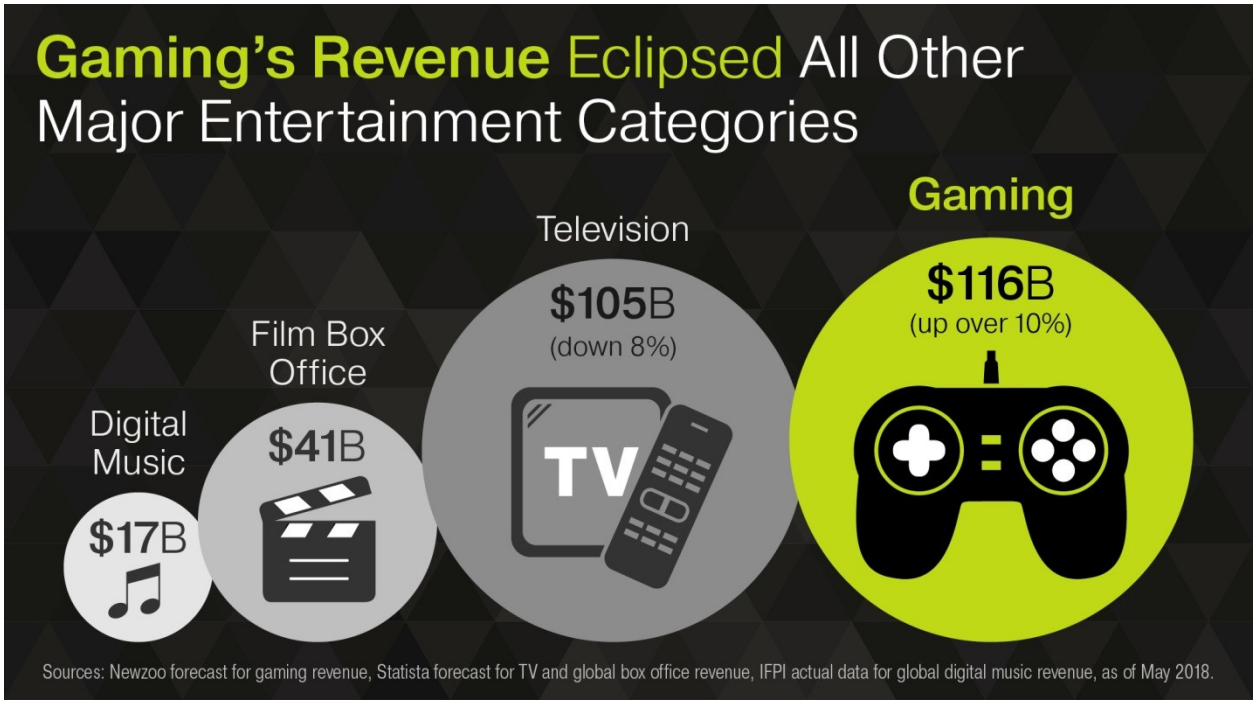

Figure 1 Entertainment / Media Revenue and Gaming - 2017 View

According to NewZoo 2019 Global Games Report (see ref[18]), the Gaming revenue is expected to be around $200B by 2022, as illustrated below –

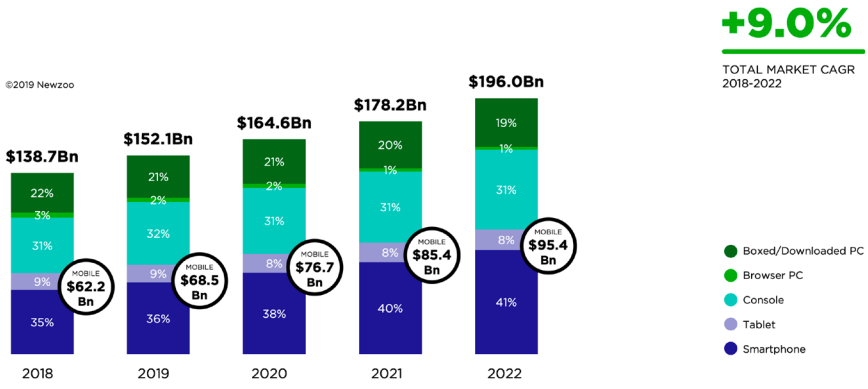

Figure 2 Gaming Revenue Growth Y-o-Y and Device Distribution

When researching this area, it's important to filter out papers and presentations that are more than a couple of years old, because of the advances that have been made in game development. In other words, it is safe to ignore papers on "Quake III" performance that date back more than 5 years ago. They do have value, but they don't represent the state of the art.

1. Classification of Games

There are different types of video games, categorized by their characteristics or underlying objectives. Game developers and publishers categorize their game titles accordingly. The popular categories are listed below, please see the complete listing here (ref[11]):

- Strategy
- Action
- Adventure
- Sports
- Simulation
- Board

The categories or genres can also have subgenres, and games could fit into multiple genres! For example, Action genre has multiple subgenres – platform, shooter, fighting, stealth etc.

In many respect, MMORPGs (Massive Multiplayer Online Role-Playing Games) and Action/FPS (First-Person-Shooting) are among the most popular online games, and now attract millions of users who play in an evolving virtual world simultaneously over the Internet.

Suffice to say, not all genres require the same network performance. Action oriented games, for example, would require a lot more strict network performance (latency, bandwidth etc.) than Board games or strategy games. The latter categories do not require strict latency/jitter treatments from the network. Figure 3 Gamer statistics. source - https://www.theesa.com shows the types of games and devices consumers use:

Figure 3 Gamer statistics. source - https://www.theesa.com

This paper focuses on multi-player shooter type games because those are:

(a) Among the most popular.

(b) They require the strictest network performance.

There are of course many other types of games, and even the multi-player games may have a standalone mode with AI (artificial intelligence) avatars instead of human competitors.

Another type of multiplayer game is the live-arena gamer, e.g. the live-arena built by Comcast (ref [1]) or other venues that host live games and are becoming popular. However, these are connected with a local network and are not the focus of this paper either.

2. Game Downloads

As a sidenote on network load, the game downloads can put stress on the network as well. Most games are sold via download now, and some games are huge. Call of Duty: Black Ops 3 is 101GB, and Grand Theft Auto V is 65GB. In contrast, an hour of 4K video on Netflix is about 7GB per hour, making a Call of Duty download equivalent to watching over 14 hours of 4K video!

For these game downloads, the network does play a role, but its limited to downstream speed (in bps), since new editions of the games could take a lot longer depending on the speed. While standard caching architectures can possibly handle these game updates well, and those are not the focus of the paper, it is important to highlight that game downloads can affect ISP network utilization far worse than anything else out on the internet – Windows downloads (~3.5GB in case of Win10), iOS downloads (~2GB in case of iOS 12), macOS downloads (~6GB for mohave), 4k movie download (7GB per hour) etc.

Also note that games (similar to mobile apps) get updated frequently and these updates could be multiple of GBs each (e.g. League of Legends minor update in Sept'2019 was ~2GB). In other words, just a minor update of a game could be more than the entire mobile/laptop OS download.

> Imagine the network contention that may arise if 50 users in a domain are downloading the latest game edition, while 10 users in the same domain are playing the game. In a given network domain, this may even cause a congestion.

Network QoS should be considered to deal with any potential network contention when new games or new versions comes out.

3. The Roles of the Game Server

Some assume that the game server acts as a directory used for the initial game setup and that following the initial setup the gamers play peer-to-peer. This is not entirely correct. As described in the section 5 ("lag compensation"), the game servers play a critical role in processing the "world view" messages that the clients send them and gamers connect to the server, not to each other (for security and privacy reasons as well).

The game server provide "match making" to pair up gamers correctly. For some game companies, the match making algorithm is a closely guarded secret, however generally speaking, these two criteria play a role in the matching:

- Skill level: game companies want to keep gamers engaged. If they pair a beginner with an expert, then the beginner is likely to be eliminated quickly, get frustrated and stop playing the game. Such a pairing is not enjoyable for the experienced player either.

- Latency: By pairing gamers with similar latency, it's easier to place everyone on the same timeline and reduce instance of "shooting behind the corner" (see more in section 5).

For effective match making, its best to have access to as many players as possible. This means that many massive multiplayer online (MMOs) gaming providers are unlikely to place their game servers close to the clients. There is a price to be paid for that flexibility, as we discuss in peering (section **Error! Reference source not found.**).

Some games can benefit from placing a server closer to the user, in particular, games that are geo-local such as "Pokémon GO", however, these are not the focus of this paper.

4. How Gamers Perceive "Lag" vs. How Network Engineering View It

Gamers often talk about "lag" when their avatar is not responding well to commands or does not move smoothly. Some assume lag is the same as latency, but it's not. "Lag" is a measure of game experience, and all of the following network impairments may result in what a gamer perceives as lag:

- Packet drops
- Packet jitter
- Packet latency

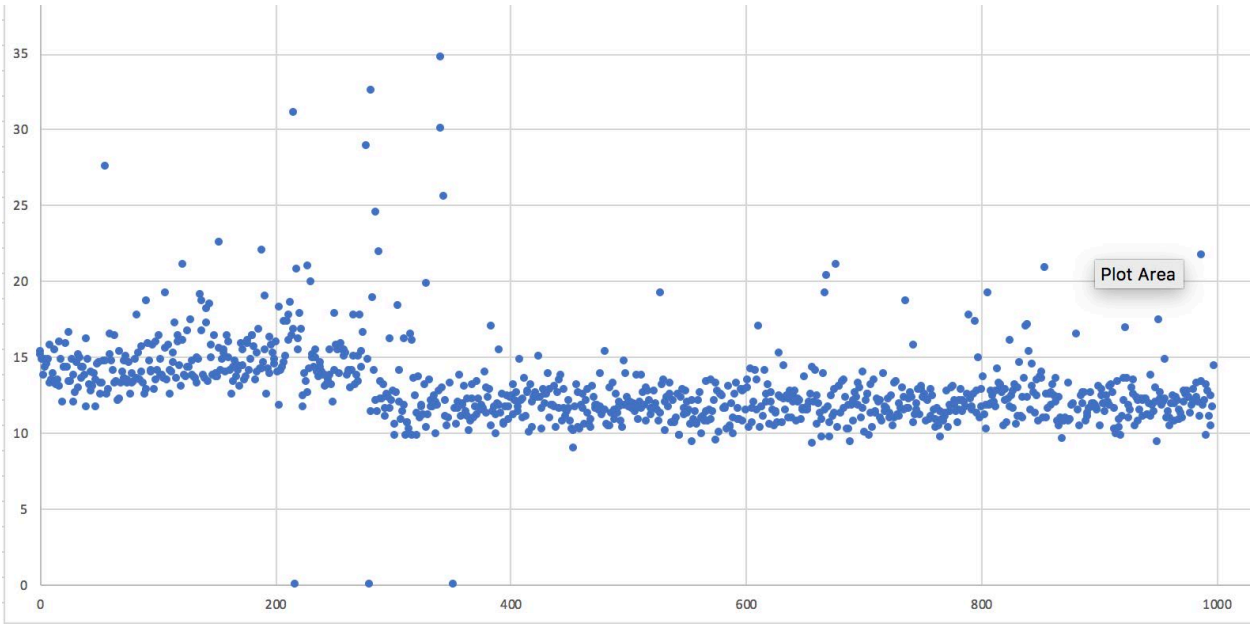

Figure 4 A 1000 pings jitter sample

To get a sense of what "jitter" means let's take a look at Figure 4 which a sample of 1000 pings over a cable network. It's meant illustrate what "jitter" looks like and how it can impact games without getting too much into the details of the reasons this particular sample looks the way it does. In a perfect world the ping time would be constant, but here we see its hovering around 10ms-15ms. As long as the ping time remains in this range one would expect a smooth playing experience. However, each jump over 30ms may represent an instant where the gamer experienced "lag".

As we explore in section 5, a stable latency is the least damaging for game play because lag compensation can deal with it well. It is packet drops and packet jitter that are the most damaging, therefor an obvious first step for improving gaming experience would be to make sure that the gamer's upstream and downstream connections are clean before deploying more advanced techniques to improve gameplay.

5. Lag Compensation

Let's consider the simplest case of two gamers, each with a 10ms delay to the game server. For our discussion we can assume no packet drops and no jitter. We would obviously love to have no latency, but that's physically impossible: the images are rendered 30 or 60 times a second (33ms or 16ms delay), packet forwarding over the network is limited by the speed of light (in practice, speed of light is 1/3 slower in optical fiber (silica glass) based network linkslight), routing hops and more. But for our example, we stick to our 10ms figure. Here is the basic issue:

- Player A sees an image of player B and shoots at it.
- In reality what player A sees is player B position 20ms in the past
- What if player B is not in the path of the bullet anymore?

In order to know whether or not to register a hit, the server uses a set of algorithms called "lag compensation". The general idea is to establish a common timeline in order to decide who ended up hitting who. There are three timelines to consider: shooter timeline, target timeline and server timeline. In general, preference is given to the shooter's point of view, most likely because if one sees the target avatar in their cross-hairs when they pull the trigger, then a hit is expected. A good discussion of lag compensation is covered in a videoclip from Blizzard in ref [3].

In a real network with packet loss and jitter, some movements need to be predicted because the player position updates may be lost or jittered too much and in any case are not synchronized with the frame rate. In such a case the players position has to be assumed and later verified. If the prediction does not work well, it will result in an artifact called "rubber-banding" where an avatar appears to be teleporting back and forth because its assumed position is replaced by an updated one that is significantly different (if the prediction was good, then the update would not change much and gameplay would be smooth). Similar prediction algorithms are deployed on the client side as well to make movement appear smooth with the same risk of rubber-banding that may occur every once in a while.

The lag compensation algorithm highlights a problem that occurs with any distributed system and is described by the "*consistency*, *availability*, and *partition tolerance* (CAP) theorem" (see reference [6]). Simply put, the CAP theorem proves that it's not possible to have a distributed system that is fault tolerant, consistent and high performing all at the same time. However, it is possible if two of the three are chosen. In gaming, the system is desired to be high performance (meaning low latency) and fault tolerant (meaning surviving packet drops and jitter) and so have to resort to "eventual consistently" – and that means prediction and lag compensation.

The combination of prediction and lag compensation can result in an artifact called "shoot-behind-the-corner" that may actually give an **advantage to the player with the higher latency**. The way it works is that the player with the higher latency can hide behind the corner, shoot a target and hide again. The target player may never see the attacking player. There is even a hacking tool called "premium lag" (https://premiumlag.com) that advertises the following: "A lag switch works by cutting off your outgoing data without cutting your incoming data. When you hold the button, you are essentially off the radar and your character appears frozen to other players." Because of the way the lag compensation works, once the lag switch is on again, the hits would be registered with the server.

Having said that, it is important to note that lag compensation makes sure that the shot was accurate but does not change who shot first. Assuming both players shot correctly, the one that shot first still has an advantage.

Table 1 provides a small sample of game engines used in multi-player games, though there is a surprisingly large number of these game engines. Most game engines employ lag compensation algorithm(s) so as to normalize server-side state (of the game) for each player as that player's user commands are executed. See ref[19] and [20] for detailed insight. Interestingly, each one of them might react to network conditions differently, as well as deploy different optimizations to deal with Lag Compensation, however, the basic issues that we outlined are fundamental to any distributed system and the appropriate solution/s need to be implemented on any game engine.

Table 1 Game Engine Examples

Game Engine	Sample game
Unreal 4.0	Fornite
Unity	Rust
CS;GO	Source
AnvilNext	Rainbow 6
Riot Games Engine	League of legends
MT framework	Monster hunter

The key take away from all the above is that **latency is not the primary experience killer for games, rather jitter/drops are,** because lag compensation can deal with fixed latency quite well (up to a point), and that's in addition to the "match making" that pairs players with similar latency. On the other hand, it is packet drops and jitter that are more directly related to what gamers experience as "lag" (avatar not responding or rubber-banding) because jitter/drops are the artifacts that cause lag compensation to miss on its prediction algorithms.

6. Human Response Time

When discussing latency, it's good to account for the human response time in the overall equation. By response time, we are referring to how quickly users react to changes they see on the screen. A fun little exercise is to measure response time using the website listed in reference [8]. Note that the website runs the app directly on the browser so there is no network latency in this measurement. The website states that a slow PC/browser can add 10ms-50ms to the measurement:

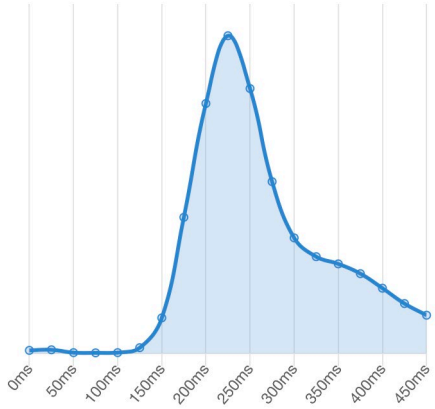

Figure 5 Human Response Time

The website states that from the statistics collected across 80 million clicks, the median reaction time is 273 milliseconds. The average reaction time is 284 milliseconds.

This means that small latency differences, in the range of 10ms or so, are in the statistical noise of human response time and do not give a significant advantage. Having said that, two issues need to be taken into account:

- There is a mismatch between the times it takes us to detect a change vs. how fast we can respond to the change : while response time is in the 250ms range, the human eye does perceive movement at 30-60 frames per second (33ms/16ms). Therefore, smaller latency makes the lag compensation algorithm work better and as a result, provide a more consistent and smoother gameplay experience.

- Delay/jitter can be additive, so reducing the delay in small amounts each domain (CM, home gateway, WAN, etc) can help for the end-to-end delay budget.

7. How is the Network Performance Profiled?

Game developers use ping from the client to the game server in order to estimate RTT and report ping times for all participants of a game. This means that if one uses fancy classification techniques to divert gaming traffic to a dedicated service flow it might not be detected by the game engine if the pings go on a default traffic path.

How are ping times related to gameplay ? Generally speaking, gamers tend to align ping times around a 30ms quanta that corresponds to a 30 frames-per-second updated.

- Excellent latency: anything below 30ms
- Acceptable latency: anything between 30ms-60ms
- Playable latency: 60ms-90ms
- Bad: 90ms and above, though depending on the game and lag compensation algorithms as much as 150ms is deemed “acceptable”

Looking into source code of Unreal 4.0 (see ref [7], Unreal 4.0 is the open source game engine for games such as Fortnite), one can see that that particular game engine refers to ping times as “QoS” - which is quite different from what network engineers refer to as QoS. See (ref[11]) for the latter.

Given the impact that drop/jitter have on gameplay, the reliance on ping as a performance metric is painting an incomplete picture, though it’s understandable that game developers want to reduce the complex issue of network performance to a single well-understood number. One should note that to some extent, ping time is a “marketing” number – the same way that comparing CPUs based on MHz rating paints only a partial number of a CPU performance, but at the same time is an easy to understand figure.

Figure 6 is a capture of a CS;GO scoreboard (based on the “source” game engine). Note that the “ping time” is a column in the scoreboard (client to server) and using a special debug command, a user can view their overall ping time (roundtrip).

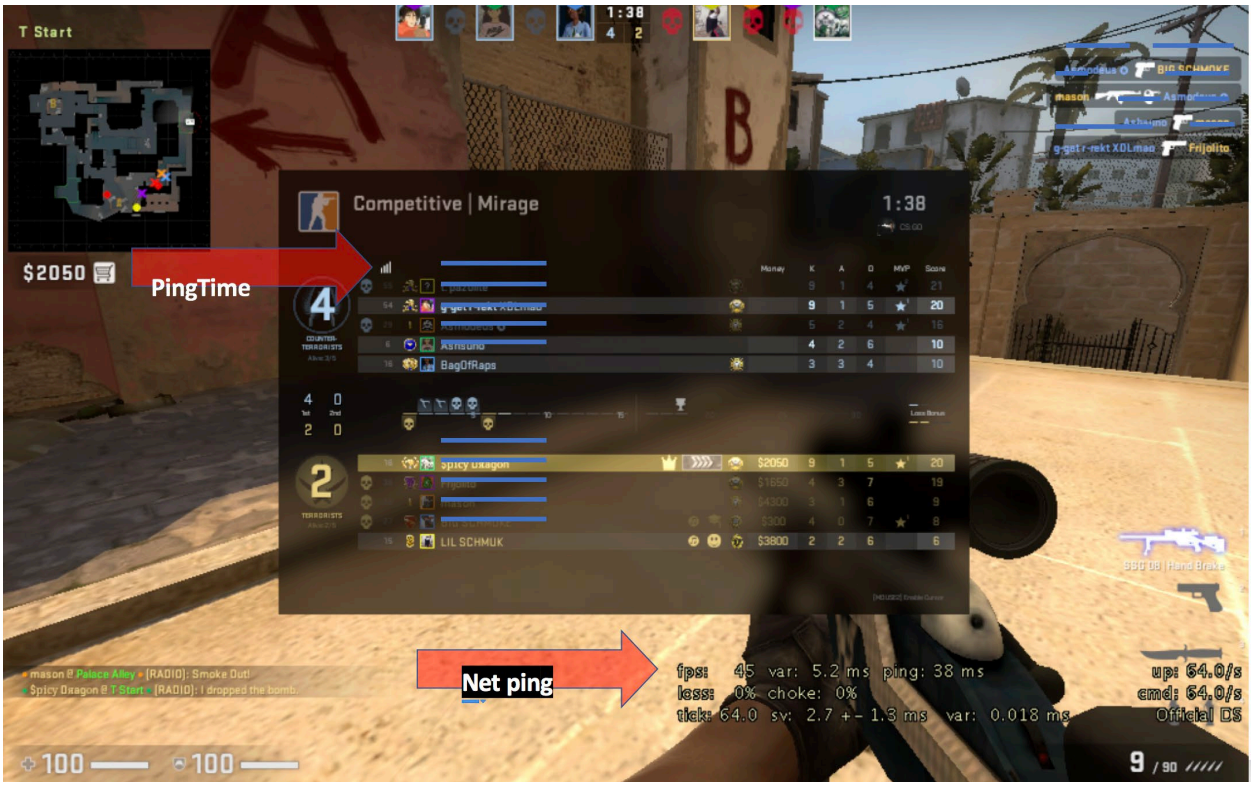

Figure 6 CS;GO scoreboard

8. Packet Transport

Gaming traffic is mostly UDP – whether video or audio.

Game engines use UDP for transporting "world view" coordinates between clients and servers. TCP does not make sense for games because:

a. Games do not need bulk transfers so all of TCPs abilities to sense the network capacity and optimize transfer rates are not useful at all for gaming.

b. Games don't need a reliable transport. If a player coordinates are lost, then it makes more sense to transmit more recent update (and rely on prediction by the lag compensation algorithm) than to re-transmit the old coordinates.

In addition to the video part of the game, there is a separate channel for voice which is UDP as well. Transfers of game updates are of course bulk transfers and use TCP.

Note that Cloud based Game streaming (see section 20) does not use TCP either, but a specific UDP based streaming protocol (originally developed by Google, now getting standardized as HTTP3.0 by the IETF) and WebRTC extensions.

9. Gaming Traffic Characteristics – Packet Sizes

Gaming traffic flows can have many of the following characteristics, depending on the game genre/sub-genre and depending on whether console or PC or mobile or Cloud delivered. For online games, the following characteristics are typical (yet to be confirmed for Cloud gaming such as Stadia):

- Long lived flows
- High packet rate
- Small and regular packet sizes
- Fairly regular packet inter-arrival times
- Bandwidth usage (Low to High depending on the game)

Note – League of Legends (one of popular games) comprises constant bitrate with small packet sizes (~55Bytes). However, other games such as the XBOX battlefield can have packet sizes around ~700Bytes.

It is interesting to note that downstream:upstream ratio is proportional to the number of players. For example, in case of 16 player game, the downstream:upstream traffic ratio could be close to 16:1. In case of 64 player game, the ratio could be close to 64:1. The reason is that the higher the number of players, a lot more info (changes) for server (since it has to aggregate all the clients' actions) to send to each player's device.

Domains & Recommendations

End-to-end game performance is a multi-domain problem. This section will outline the domains involved in game performance. The delay/jitter/drops that we discuss in the following section is additive, and in some cases even if one domain still meets reasonable performance criteria, the combined effect of several domains can be significant.

10. Game Rendering

The compute resources needed to render a game are not the focus of this paper, but they are part of the overall game experience so for completeness we will overview them.

The tradeoff a gamer needs to make is cost of the hardware vs. the quality of the game animation. If a computer does not have a powerful enough CPU, GPU and memory then either the frame-rate, or resolution or both have to be reduced to keep gameplay responsive. If latency is caused by lack of computing horsepower on the client hardware, then there is nothing that lag compensation can do about it.

It is a little easier to manage the quality/responsiveness tradeoff in console games, as opposed to PC games, because the game can be optimized for a specific platform (the console).

Another issue with PC games is that other applications might be running in the background, and a periodic software update or system backup might start in the middle of a game. While easy to fix, these are some of problems that we have to add to the list of things to debug when trying to assure end-to-end game experience and are clearly outside the scope of the Service Provider world.

It's worth noting that most Gamers playing action type games are savvy consumers and can usually resolve the issues that are within their control on their own. However, other types of gamers would not be as savvy and may expect gaming provider or Service Provider to look after their needs and assurance.

11. Home Network Domain

The home network is one of the more hostile environments for gamers. WiFi can incur delays of 40ms and have a large percent of packet drops. It is true that serious gamers are recommended to use wired connectivity (i.e. connect the gaming device directly to their home router using an ethernet cable), but in reality, many don't.

In addition, if the home gamer shares the connection with other people in the same household they may run into congestion and buffer-bloat issues in the home (before even getting to the cable modems). Some companies offer home gateways that can mitigate these issues and those seem to be working to reduce ping times and improve game performance. These solutions have built in buffer-bloat management algorithms.

As a side note, one should not confuse the home gateway and a cable modem. The cable modem handles the DOCSIS protocol, which is basically the interface between the CMTS and the CM, while the home gateway typically performs functions such as NAT and WiFi access. Even in cases where both the CM and the home gateway are packaged in the same enclosure, we should still treat them as functionally different entities.

All the above is to make a clear distinction between applying buffer-bloat mitigation at the home gateway vs. doing the same at the cable modem as we will explore in section 13.2.

RECOMMENDATIONS:

1. Prefer Wired connectivity to WiFi, if possible
2. Prefer 802.11ax (or 802.11ac) if using WiFi connectivity, given the scheduled mode support
3. Prefer 5Ghz or higher (more number of non-overlapping and wider channels, lower contention); However, resort to 2.4Ghz non-overlapping channels (1,6,11), if possible, in case of having brick walls between AP and gaming device.
4. Use routers that avoid bufferbloat issue
5. Disable unused WiFi modes (such as 802.11a,b,g,n).

6. Consider dual-channel WiFi that can separate out Tx and Rx on different channels. See more details on cablelabs' work in this area (reference [16]).

7. Avoid Mesh WiFi solutions, given the latency impact

11.1. WiFi – WiFi5 vs WiFi6

Home network commonly employs (802.11 standards based) wireless connectivity to access the network and consume services/content, given the sheer convenience factor. Interestingly, 802.11 standards have evolved quite a bit over 2 decades :

- 802.11b (WiFi1*), released in 1999
- 802.11a (WiFi2*), released in 1999
- 802.11g (WiFi3*), released in 2003
- 802.11n (WiFi4), released in 2009
- 802.11ac (WiFi5), released in 2014
- 802.11ax (WiFi6), being released in 2019

*Not official naming,

However, despite the evolution, 802.11 wireless connectivity at home has remained somewhat problematic for superior gaming experience in certain cases, due to indeterministic latency and drops.

A major cause of latency in WiFi is that it's a carrier sense system and when many end stations compete for bandwidth, it can experience low utilization and high latency.

WiFi6 solves this latency problem by having a "scheduled" mode where it can predict bandwidth demands and schedule transmission time in advance and thereby avoiding issues with congestion and carrier sensing.

How does a WiFi6 detect an application need for bandwidth ? Well, the standard itself does not provide any means, but it can be as simple as setting fixed transmission time for all packets directed to a specific game server and by doing that assuring a dedicate channel/slot for game traffic. This can be one of the "advanced" options in the WiFi6 router settings and something that most serious gamers should be savvy enough to do.

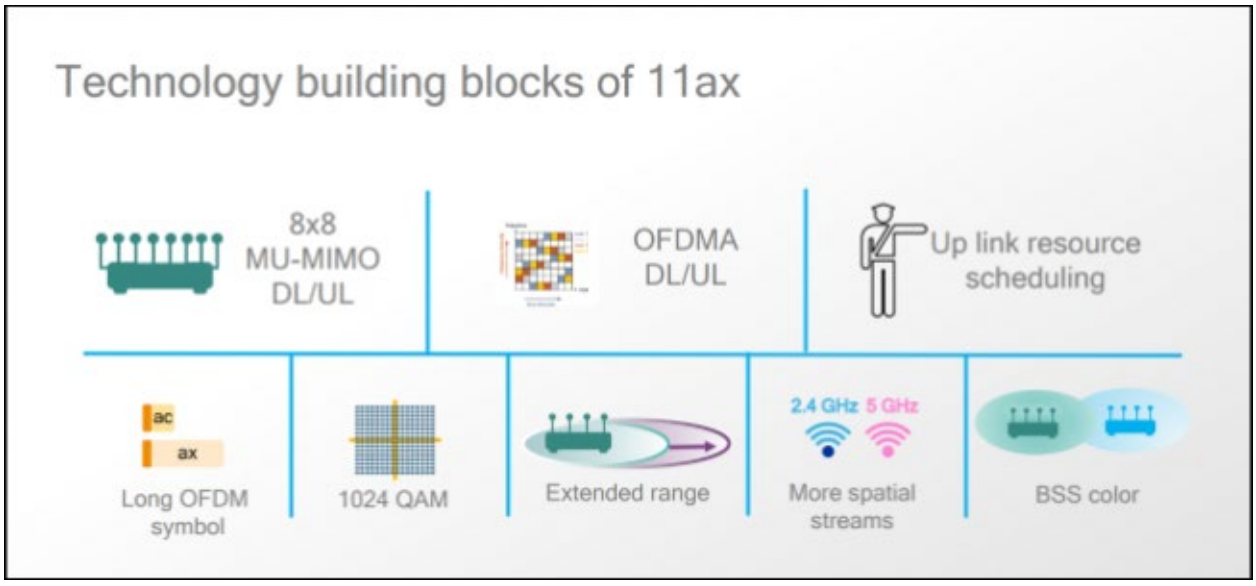

Figure 7 Wifi Building blocks

Figure 7 depicts the base WiFi6 building blocks, see ref [13]

12. SP Network Domain – First Mile

First mile is the lag of the network that connects a home to the first active outside the home. Since this paper is written for a cable conference, we will focus on the DOCSIS drop/jitter/latency. As discussed in section 5, it is the drop and jitter that damage game experience more than the fixed latency.

Packet drops in cable networks are typically caused by RF impairments, and the first phase for improving a game experience would be to make sure the RF part of the plant performs well. Remember that lag compensation can mask packet drops up to a point and if it happens to predict well, the game experience might still be reasonable, else artifacts such as “rubber-banding” will creep up.

Jitter in cable networks could occur because of contention slots. In order to send a request, the CM has to first contend for a “DOCSIS contention slot” to send the request. The time it takes to acquire a contention slot is random, but it gets worse, as the utilization gets heavier. In bulk transfers, the impact of contention is secondary because the DOCSIS protocol allows additional bandwidth requests to be piggybacked, however in short transactional transfers, such as the ones used to transfer a game “world view”, the contention channel is the primary method for sending bandwidth requests.

In principle, buffer bloat and congestion on the access could cause jitter as well, but it’s a topic of further study whether the above manifests itself as a relatively stable delay or as jitter.

RECOMMENDATIONS:

1. Keep the RF part of the cable plant clean
2. A dedicated DOCSIS service flow would assure game traffic prioritization and easier debug.
3. Follow some or all of the Low Latency DOCSIS recommendations (at least reduced map times and proactive scheduling).
4. Reduce congestion, either by increasing upstream rates or node splits

13. Achieving Low Latency on a DOCSIS Network

When analyzing any system for delay the following contributing factors need to be considered:

- Propagation delay: typically, the speed of light, or in fiber 2/3 of the speed of light
- Transmission delay (serialization/encoding): the time it takes to send the bytes to the wire (e.g. serializing a 1500 bytes packet on a 1Gpbs link will take 12 microseconds)
- Processing delay (media acquisition): any computation time it takes to process the packet, for example, calculating a CRC.
- Queueing delay: once the packet is queued for transmission how long it may end up waiting in the queue.

13.1. Sources of Delay in a DOCSIS Network

Analyzing all the delay elements in DOCSIS could be an SCTE paper in its own right (and papers have been written on the topic, see ref [14]), so we will keep the discussion at a high-level, ignore the 2nd order artifacts and focus on the elements that are most relevant when comparing DOCSIS to other technologies (e.g. PON).

DOCSIS Media Acquisition Delay: Because DOCSIS is a request/grant/data system, the propagation delay is multiplied by 3:

- The time it takes the request to propagate in the upstream direction,
- Then the grant to propagate in the downstream direction
- Followed by the actual packet in the upstream direction.

DOCSIS 3.0 made a significant improvement because it provides the ability to “pipeline” the requests. So, for a large file-transfer, the request/grant delay becomes relatively negligible. For example, if it takes 200ms to load a web page, then an initial delay of 5ms is not that significant. To illustrate what pipelining means, imagine the following scenario: say a person goes grocery shopping in a store that is a 10-minute drive from their home. In the pre-DOCSIS 3.0 case every item purchased would require a separate trip. Buying 5 apples would require 5 trips and a total roundtrip of 10x2x5 = 100 minutes.

And here comes the interesting part of the analogy…one might expect that DOCSIS 3.0 is analogous to packing the 5 apples into the car and finishing the transfer in one roundtrip, but a

better analogy for DOCSIS 3.0 would be sending 5 cars at the same time, each carrying a single apple. The result however is the same, all the groceries would arrive in 20 minutes (roundtrip time). The above holds true for longer bulk transfers, but for short transactions, the delay is additive since there is no "pipelining" and short transactions are relevant to gaming. Back to our example, if you were really hungry for an apple you would not care that after 20 minutes you could get 5 apples, you just want one as quickly as possible.

DOCSIS Propagation Delay: as transmission rates over cable get faster the transmission delay is in the range of microseconds and not a significant contributor. Having said that the transmission over the fiber part of the network can be in the range of milliseconds depending on the length of the fiber part of the cable plant.

DOCSIS Media Acquisition Delay: the most significant contributor to processing delay is the downstream interweaver, usually in the range on 2ms or so. It can be made shorter but at the risk or reducing the fidelity of the transmission.

DOCSIS Queuing Delay: Queuing is how long a packet waits in a queue before it is served. DOCSIS has tools to deal with queueing delay by sorting packets to flows so that the high-priority flows get services quickly before they form a long queue (analogous to the faster boarding service for business class in an airport). In principle, gaming packets could be classified to a dedicated high-priority service flow to assure that queuing delay is minimized. Another option is the Dual-Queue part of LLD (low latency DOCSIS) which we will explore in more detail in the following sections.

For more detail please see reference [14].

13.2. CableLabs LLD

In order to reduce latency over the cable plant, CableLabs has initiated the development of "low latency DOCSIS" (LLD). LLD is described in detail in reference [14], for our paper we will give a high-level overview of the two features LLD enables:

- *Proactive scheduling*: The simplest way to reduce the request-grant delay is to eliminate the need to send a request. Proactive scheduler is sending a stream of grants even if the modem is not requesting. It is like UGS (unsolicited grant service) but for data. It is a way to trade off bandwidth (since some of the unsolicited grants may be unused) for lower latency. It's worth noting that the mobile 5G standards advocate a similar technique to facilitate low latency over mobile, and similar technique is used in WiFi6. A way to mitigate the bandwidth loss due to unused grant is called "grant sharing" where a grant for service flow A can still be used for service flow B if service flow A happened to not use it.

- *Dual queue*: TCP traffic causes buffer buildup because the way TCP works is by sending as much data as it can until it notices packet drops. This is a good method for bulk transfers because it keeps the network full, but it triggers a phenomenon called "buffer bloat" which can cause excessive delay. The dual queue in LLD is a collection of methods to help keep buffers shallow and at the same time assure that "well behaved"

traffic is rewarded and experiences less queueing delay. For a detailed description of Dual Queue see reference [TBD]

- *More frequent MAPs*: The typical MAP interval for many CMTS implementations is 2ms. This means that worst case the CMTS might have processed a request but has to wait 2ms before sending this grant in a map message. LLD specifies a preference for 1ms MAPs. By doing that it reduces the MAP wait time to 1ms.

13.3. DOCSIS Delay vs. PON delay

Both DOCSIS and PON (EPON/GPON) are point to multi-point technologies and are similar on the need for a request/grant/data cycles (even though these are called differently in PON it's a similar concept). So, why are the PON vendors marketing their networks as "low latency" ? Two key differences are:

- *No Contention in PON*: think of PON as RTPS (real-time polling service) only. There is no option to contend for bandwidth and all slots to carry bandwidth requests are pre-scheduled and dedicated to the end station.

- *Shorter processing delay*: there is no "downstream interleaving" and other physical layer related delays in PON because currently its only 0's and 1's with no modulation (aka "baseband"). This already cuts about 2ms from the delay

- *Immediate grants*: There are no periodic MAPs in PON. Every request can be granted individually so there is no MAP wait time. This can cut another 1ms-2ms.

- *Lower number of subscribers and higher bandwidth*: This is not a protocol advantage, but because PON cannot be split to a large number of subscribers (being "passive" it loses half the power for every split) we have a smaller number of subscribers and more bandwidth for each Service Group then what is typical for cable.

As we explored in this paper, we have several tools to reduce the DOCSIS delay getting within a few milliseconds of what PON can achieve. Furthermore, as is evident from our discussions on lag compensation, frame-rates and human response time, it is clear that a difference of single digit milliseconds is a non-issue for gameplay.

14. SP Network Domain – Second Mile

In most networks, the "2nd mile" is the network lag that connects the aggregation point to the internet exchange (or peering point). In most networks the 2nd mile is not over-subscribed and typically not a source of congestion, jitter or drops. However, it may be important to prioritize gaming UDP traffic in case of congestions resulting from network link/node failures etc.

RECOMMENDATION

- Keep it simple and keep the 2nd mile rich with bandwidth, monitor performance and buffer depth in the routers/switches that make the 2nd mile.
- Mark UDP gaming packets with a higher DSCP code point.

15. Internet Domain

This is an area that is an unexpected source of pain in terms of latency and jitter. Here is why:

- The game server usually are centralized, so that it has the most flexibility for matching players (with the exception of localized games such as "Pokémon GO"). Some game severs have been located in the center of a continent to have equal distance from most gamers.
- Centralizing a server is not necessarily a problem, but because of the way the Internet is built it becomes a problem. What we call "Internet" is not the simple network cloud that we draw, but rather a collection of smaller networks, transports, peering points and service provider networks. Routing IP packets between all these networks is based on minimal cost does not always equate to minimal latency. Reference [4] includes an excellent discussion showing how Riot Games built their own network in order to avoid routes that are minimal cost optimized but not latency optimized. Even at fiber speeds, these inefficient routes can add up to 10's of milliseconds of delay.
- The interesting challenge is that there isn't any big Gaming traffic Aggregators (similar to how content aggregators mushroomed a decade ago or so), nor any dedicated gaming exchange. Perhaps, an opportunity for Internet Exchange Points (IXPs) to become Gaming Exchange Points (GXPs). Else, ISPs and Gaming Providers will need to have direct or indirect peering that is latency optimized.

In general, it is beneficial for ISPs to have dedicated peering, if possible, to offload gaming traffic, whether via the gaming exchanges or via direct connectivity, to the networks having the optimal routing (latency, hops etc.) to the data centers hosting the gaming servers.

There are several strategies and several companies with products that can help with the above. They generally fall into one of three categories:

1. *VPN solutions*: by encapsulating game traffic into a VPN it's possible to steer it into the best peering point and may help divert traffic to a good path.
2. *Overlay solutions:* these solutions take advantage of the fact that the Internet is more than a simple "network cloud". There are many data centers in various places in the Internet and traffic can be routed from one data center to another thereby creating an overlay that effectively overrides packet routing decisions based on "minimal cost".

3. *Custom-built networks and custom-built edge solutions*: some game houses host the game servers on public cloud solution and these public cloud solutions can have their own network optimizations. In other cases the game provider will create its own peering network (see reference [5] presentation Blizzard networking solution as well as reference [4] already mentioned).

RECOMMENDATIONS

- Use one of the solutions mentioned above to optimize the IP traffic paths between the game server and end devices.
- Seek and Implement direct and optimized peering relationships with gaming providers.

16. Data Center Domain

The game servers are hosted in a cloud, either public or private, and cloud applications are subject to delay/drop/jitter like any application.

The data center is a dynamic environment where compute loads can be moved at any point and each such move can cause delay/jitter/drops on the server side. A private data center may be easier to control, and even in a public data center it is possible to pay more to get a higher service assurance but clearly there is cost to both options.

RECOMMENDATION

- Deploy an application monitoring system in the data center to track how well the application is running.
- Perform Cost analysis of server performances vs. data center options (private/public/higher service tier)

Gaming as a Managed Service

Establishing a dedicated game connection can be viewed as similar to connectivity services sold to business subscribers. The basic tools that DOCSIS provides to build such a connection are the packet classification rules, DOCSIS service flows, and a policy framework. The following section will go into the detail of creating a “game connection”.

Note that if game packets are directed to a dedicated DOCSIS service flow, then there is little need for active queue management (such as dual queue which is part of LLD) because there will be no buffer bloat in a queue that serves only the gaming data. In addition to that, a dedicated service flow can use RTPS/nRTPS or dedicated predictive scheduling which will further reduce jitter because no contention slots will be used.

17. Classification

Game traffic needs to be classified in order to be directed to a DOCSIS service flow. As we discussed in section 11, the home gateway is a separate entity then DOCSIS and in the home gateway a user may directly program any packet with a destination of a game server to have "special treatment" but there is no such option for the cable modem, so how can it be done ?

PCMM (Packet Cable MultiMedia) is a solution where a user application can request QoS for the DOCSIS network. Having said that, PCMM was not a commercial success and we use PCMM as an example for a policy framework. In the solution outlined below, PCMM is meant as a reference that's understood by cable operators. However, a wireless policy framework such as PCRF can be used just as well.

The proposal below puts emphasis on game server to cable network interaction as opposed to game client to cable network. It may be intuitive to have the client send a request for a dedicated service flow, but in this proposal, the game server is the one to signal the exact classifier for the following reasons:

- It's easier to establish trust between the game server and a policy server than between the home client and the policy sever. This is for two reasons : (a) the game company already establishes trust with clients, filtering out known abusers, so a request coming on behalf of a client is already sanitized, and (b) easier to build trust with a small number of business entities on the server side then with a huge number of home clients.

- The server sees UDP packets post-NAT. The client side can report packet pre-NAT. This way, we don't have to deal with NAT issues since we have the full picture (when we view it from the server side)

The following ladder diagram shows how this proposal can work with PCMM; but it's only a reference so we can have a concrete example. In this day and age of micro-services, it's easy to build a function that just sends common open policy serive (COPS) messages and has a different policy framework (possibly a PCF/PCRF) on top of it.

This proposal does require some changes to the game client. By way of inspecting the open source code for Ureal 4.0 (one for the most popular game engine, which happens to be open source) it seems possible.

Figure 8 depicts a proposed exchange between a policy framework (PCMM in this example, based on reference [9]):

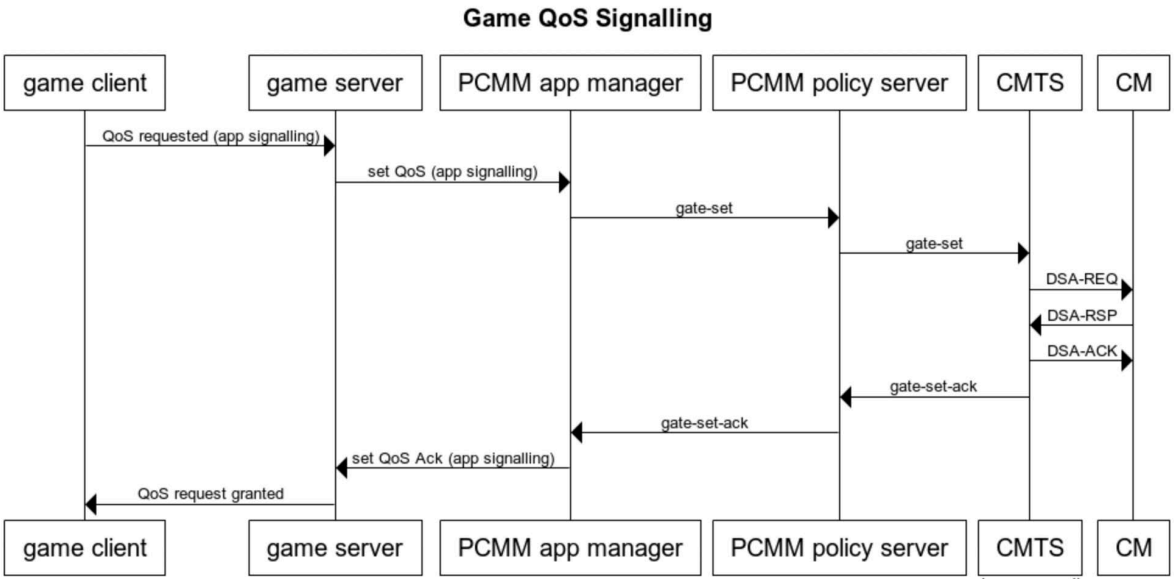

Figure 8 Game Signaling

The initial signaling is between the game client and the game server. In fact, it can be part of the normal announcement that a gamer is joining the game and not a dedicated message.

The game server is in the best position to validate that the client joining is not a bot or was blacklisted for any reasons (such as cheating or flagged by other gamers). If the client is approved, the game server will signal the PCMM application manager a request for QoS, and the application manager will forward the request to a policy server for approval, this is all in accordance with the PCMM specifications.

From this point on we are dealing with the DOCSIS domain and a standard DOCSIS request for creating a service flow. We are focusing on the "success" case where QoS is allocated and a positive indication is propagated all the way back to the game client.

As outlined in Section 7, if the game is using pings to measure network performance it is desirable to classify pings to the same flow as the actual gaming traffic.

Since the flow created is designed to carry only game traffic, there should be no need for queue management because the game traffic itself, being UDP based and at fixed interval, will not cause buffer bloat.

18. Gaming as a Marketing Play

The marketing departments in any big operators are aware of the interest gamers have in "ping times" and in several cases use it as a sales tool, either to promote specific products such as "gaming package" or as a selling point against other operators. The latter is a point made by PON providers against cable providers. As we explored in this paper, the technical merits of PON against cable when it comes to latency are not significant. Howeve,r as a marketing tool, it can be effective to claim an advantage.

From a quick survey of "gaming packages" offered by cable operators, it seems like they fall into two categories in the moment:

- Higher speed services that are packaged as "better for gaming" because they reduce congestion at the home
- Collaboration with a game optimization vendor that is packaged with the cable provider offering.

19. Assuring gaming performance across domains

Once a packet is classified to a flow, it can be marked in a way that can make it recognizable as a gaming packet across domains. For example, the CMTS can place it in a particular VRF or VPN. It can also help with preventing attacks on the game because if an authentic game flow is marked, then it's possible to treat non-identified flows as "suspicious".

Defining an API between the game server and the network for assurance can help as well. The API can help isolate networking problems even if the caller of the API is not a networking expert. For example, if traffic is sorted to flows and these flows associated with a particular user, then packet counts can be compared across the path to isolate domains where packets are dropped.

Because there are many game developers and many service providers, it may call for a "middleman" between the two organizations to help define and operate these APIs.

B2B vs B2C monetization

As outlined in reference [4], game developers would invest in improving network performance to the point of building their own network. This means that instead of a B2C (business to consumer) monetization strategy of "have a customer pay X to improve gaming experience", it may make sense to have a B2B (business to business) monetization strategy with the game developer. This has the following advantages:

1. Instead of a customer paying twice, once to "improve gaming" and once to a specific game developer, there is only a single payment to a game developer. One has to think about it from the perspective of a 15-year-old asking a parent for permission to buy a game for X dollars alone as opposed to asking for X dollars for game and then an extra Y dollars to the Service Provider. Note that the monetization can be per-use of the game.
2. As outlined throughout this document, the game networking performance is a multi-domain problem and the access network is only one part of it, and in actuality might not even be the long pole in the end-to-end performance of the game. When a cable provider charges money for "game performance" it implicitly assumes responsibility for end-to-end game performance and if the game experience is bad because of issues in a different domain then the toxic reviews on Reddit will follow very quickly. It's the game developer that is in a better position to assume responsibility of the end-to-end game

performance, and for them to craft an agreement with each of the domains they cross to give them a competitive edge.

3. If the APIs are changed its easier to update the server side, than force an upgrade to the clients.

Issues pertaining to net-neutrality and specifically to the relative advantages or disadvantages of a B2B vs. B2C from a legal viewpoint are outside the scope of this paper.

Future topics

20. Cloud Game streaming

Recently, Google, Microsoft, etc. have announced their cloud game streaming platforms – Stadia, xCloud etc. respectively. Per Google, "*Our vision is to have Stadia available on all devices that stream YouTube—a truly platform-agnostic service*." Stadia has the potential to become a popular alternative to game consoles since it could make playing games as simple as watching on-demand videos by clicking a button on any device.

Stadia promises that the game could start in less than five seconds after clicking on a link: no download, no patch, no install, no updates and in many cases, no hardware required. Just streaming, taking advantage of content distribution network technologies that most cable service providers have become well accustomed to.

Suffice to say, game streaming platforms such as Google Stadia demand specific network performance – around 15GB per hour of 4K game play (which is more than 2x than watching a 4K movie). Wrt connection speed:

- 1.5Mbps+ upload speed and 10 Mbps+ download speed for 720p resolution at 60 fps.
- 1.5Mbps+ upload speed and 20 Mbps+ download speed for 1080p resolution at 60 fps.
- 1.5Mbps+ upload speed and 35 Mbps+ download speed for 4K resolution at 60 fps.
- Stable/deterministic network latency.

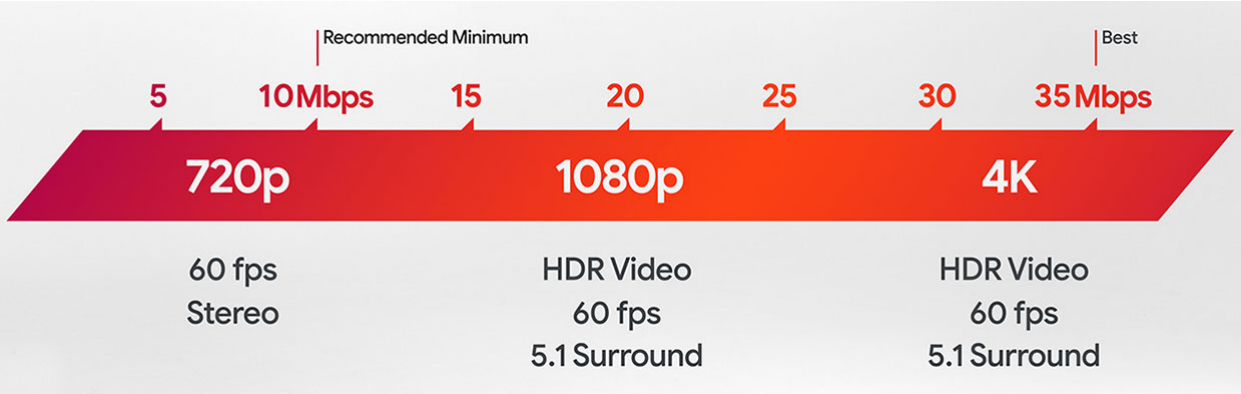

Figure 9 Stadia Bandwidth Usage from Google

The following are very interesting points gleaned so far wrt Google Stadia:

- Stadia does "game streaming" which means that instead of rendering on a local console, it renders the images in the cloud and streams them as video to the consumer. Traditional console games do not take a lot of bandwidth (since they essentially communicate coordinates, not images). In contrast, game streaming can stream tens of megabytes. It's definitely going to be a stress on the SP network if Stadia becomes successful. Some characterize Stadia as "Netflix for games".

- Stadia may leverage AI based frame-by-frame latency prediction and be able to respond to gamers faster than their eyes could perceive the responses. This may indicate a sub-13ms latency, which is pretty astounding.

- At the time of this writing, Stadia is a new service and based on announcements made in E3 it targets single player games. Therefore, many of the issues discussed in this paper are secondary for this initial phase.

- There are people who play games and there are people who watch games. More watchers than players, obviously. Google's vision is for Stadia platform to converge these two worlds together so that one can be watching a game, click and be playing a game or vice-versa.

- Stadia does require a Google controller, since the rendering is done in the cloud (around $69, relative to $300+ for an XBox). More details described at [15]

- Stadia can use the Google network, data center locations and thousands of network edges that Google manages around the globe to source the streams. Having said that, the last mile is naturally still in the SP domain.

Conclusions

The service provider is in a unique position when supporting consumer gaming. The consumer sees the SP as the "one throat to choke" for any problem that is perceived to be network problem. But in the real world, the SP controls only network domains - the access, aggregation, possibly the core and possibly a part of the home domain. In some cases, these domains might not even be the most significant culprits during a degraded gaming experience. This basic catch is only going to be exacerbated if the SP charges fees directly from the consumer for an "advanced gaming service".

How can we improve the experience for gamers and at the same time reduce the OPEX of debugging issues when the gaming experience is not good?

First of all, what can be improved in the access domain should be improved. Better RF will reduce packet drops. Dedicated gaming service flows or better queuing at the cable modem (such as LLD) will help reduce delay and jitter as well as make it easier to debug networking problems.

Secondly, collaboration with the game developers that will allow server-side probing for debugging of game issues can help reduce the OPEX required to debug experience issues. This can either be done directly or through a middleman between the service provider and the game developer. The key is the B2B relations between the game developer and service provider where it's easier to establish trust and business relations than with the B2C model on the client side.

Abbreviations

API	Application Programmatic Interface
B2B	Business to Business
B2C	Business to Consumers
PPS	Packets Per Second
BPS	Bits Per Second
PCMM	Packet Cable Multi Media
QoS	Quality of Service
QoE	Quality of Experience
VRF	Virtual Routing and Forwarding
VPN	Virtual Private Network
CCAP	Converged Cable Access Platform
CMTS	Cable Modem Termination System
LLD	Low Latency DOCSIS
NAT	Network Address Translation
COPS	Common Open Policy Service
PCRF	Policy and Charging Rules Function
EPON	Ethernet Passive Optical Network
GPON	Gigbit Passive Optical Network
IXP	Internet Exchange Point

Bibliography & References

1 Netflix shareholder report Q4 2018 (“competition” section, page 5): https://s22.q4cdn.com/959853165/files/doc_financials/quarterly_reports/2018/q4/FINAL-Q418-Shareholder-Letter.pdf
2 Comcast live arena : https://www.nbcsports.com/philadelphia/fusion/fusion-arena-become-newest-state-art-gaming-facility-philadelphia-sports-complex
3 Lag compensation explained by Blizzard: https://www.youtube.com/watch?v=vTH2ZPgYujQ
4 Riot games network: https://qz.com/790208/how-the-company-behind-league-of-legends-rebuilt-its-own-internet-backbone-so-that-its-faster-for-gamers/
5 Blizzard network @scale : https://www.facebook.com/watch/?v=2090071161265977
6 CAP theorem: http://robertgreiner.com/2014/06/cap-theorem-explained/
7 Github for QoS code in Unreal Engine 4.0: https://github.com/soxueren/EpicGames-UnrealEngine/tree/59267dc158d4e919a579a98d472fbf21bb64508b/Engine/Plugins/Online/OnlineFramework/Source/Qos - one needs a github account and an Epic games account to link to it.
8 Response time measurement : https://www.humanbenchmark.com/tests/reactiontime
9 Cablelabs PCMM https://specification-search.cablelabs.com/packetcable-multimedia-specification

10 Online Gaming Industry – Statistics & Facts https://www.statista.com/topics/1551/online-gaming/
11 Network QoS https://www.cisco.com/c/en/us/products/ios-nx-os-software/quality-of-service-qos/index.html
12 Online game viewing - https://onlinebusiness.syr.edu/blog/esports-to-compete-with-traditional-sports/
13 WiFi - https://www.howtogeek.com/368332/wi-fi-6-what%E2%80%99s-different-and-why-it-matters/
14 Low Latency DOCSIS, Greg White, SCTE EXPO 2019
15 Google Stadia https://store.google.com/product/stadia_founders_edition
16 Cablelabs Dual channel Wifi : https://www.cablelabs.com/technologies/dual-channel-wi-fi
17 Video Gaming vs Other Entertainment Revenue - https://www.gamecrate.com/statistically-video-games-are-now-most-popular-and-profitable-form-entertainment/20087
18 Global Gaming Revenue Growth - https://newzoo.com/insights/articles/the-global-games-market-will-generate-152-1-billion-in-2019-as-the-u-s-overtakes-china-as-the-biggest-market
19 Lag Compensation Algo - https://enterprisecraftsmanship.com/posts/how-i-tried-to-get-into-game-development-and-failed/
20 Lag Compensation Algo https://developer.valvesoftware.com/wiki/Source_Multiplayer_Networking

