WPI-CS-TR-23-02 Feb 2023

Measurement of Cloud-based Game Streaming Systems Competing with DASH Flows

by

Xiaokun Xu and Mark Claypool

Computer Science
Technical Report
Series

WORCESTER POLYTECHNIC INSTITUTE

Computer Science Department
100 Institute Road, Worcester, Massachusetts 01609-2280

Measurement of Cloud-based Game Streaming Systems Competing with DASH Flows

Xiaokun Xu
Worcester Polytechnic Institute
Worcester, MA
xxu11@wpi.edu

Mark Claypool
Worcester Polytechnic Institute
Worcester, MA
claypool@wpi.edu

ABSTRACT

Cloud-based game streaming lets users play games with a lightweight client by streaming the game frames as video from a server. For a good player experience, cloud-based game streaming clients need a network connection that supports the high bitrates needed for visual quality and the low latencies needed for interactivity – both a challenge when competing for network capacity with other flows. While network capacities for cloud-based game streaming have been studied, as have system responses to capacity constraints, packet losses and competing bulk-download flows, missing are comparative performance and congestion responses for cloud-based games competing with video streaming flows. This paper presents results from experiments that measure how three popular commercial cloud-based game streaming systems – Google Stadia, NVidia GeForce Now, and Amazon Luna – respond to Dynamic Adaptive Streaming over HTTP (DASH) flows on a congested network link. Analysis of bitrates, frame rates and round-trip times for the game-streaming flows and analysis of media throughput, interrupts and quality changes for the DASH flows show the three systems have markedly different responses to the arrival and departure of competing DASH traffic, with corresponding differences in the quality of the game streams and DASH videos.

1 INTRODUCTION

Cloud computing infrastructures combined with high-capacity networks have enabled the emerging market of cloud-based game systems that stream game frames as video, letting the player experience high quality graphics and gameplay with only a lightweight client. Systems that seek to capitalize on the opportunity afforded by cloud-based game streaming systems include Sony PlayStation Now, Microsoft xCloud, Google Stadia, NVidia GeForce Now, and Amazon Luna with Meta (formerly Facebook) arriving soon. Cloud-based game streaming as a market is growing rapidly with a value of $865.8 million USD in 2021, with the expectation for expansion at an annual growth rate of 48.2% from 2021 to 2027 [17].

Unlike in traditional computer games, cloud-based game streaming clients do not run full versions of the game engine. Instead, only the cloud-based server handles the relatively heavyweight game and graphics tasks – applying physics, resolving collisions, processing AI, and rending the game frames – streaming the game as video to the game client. This allows the game client to be fairly lightweight, needing only the capability to play the streamed game frames similarly to a streaming video player. However, unlike a streaming video player, the cloud-based game client sends the frequent player game input back up to the server to be acted upon in the game. This means a significant disadvantages of cloud-based game streaming is added round-trip latency from the client to the server for all player actions, and also the increased traffic required for the game frame streaming. In particular, the bitrate requirements for frequent, high-quality video frames can cause congestion, degrading player quality of experience especially in the presence of co-located network traffic.

Prior work has shown that cloud-based game streaming requires a high capacity network and is sensitive to network latency [3, 10, 22]. While studies have analyzed network traffic for specific cloud systems like Google Stadia, NVidia GeForce Now and Sony PSNow [11, 16, 38], there are only a few papers that compare aspects how cloud-based game streams respond to congestion from competing flows. This latter aspect, congestion, could be self-induced when the network capacity is insufficient to support their maximum bitrates or co-induced when the cloud-based game streaming competes for capacity with other network flows on the bottleneck link.

Previous work has compared the congestion response for some cloud-based game streaming systems competing with bulk-downloads [39, 40], whereas an alternate scenario is that of cloud-based game streaming systems competing with streaming video flows. A potentially common bottleneck situation is when a cloud-based game stream shares a bottleneck link with a Dynamic Adaptive Streaming over HTTP (DASH) flow, as might happen in a home network where one person is playing a cloud-based game while a housemate streams a YouTube or Netflix video. And this scenario is increasingly likely. Every week in February 2021, Americans streamed 143.2 billion minutes of video content [25] for an the average of 19.3 hours of video streamed per week per household[1], or about 2.75 hours day. That number increased to about 3.25 hours per day in 2022 [24]. Moreover, the bitrate requirements for video streaming service have significantly increased, as well, with more support for 4K UHD content. Live sports, including e-sports, are also streamed with greater frequency. Such streaming bitrates can be as high as 35 Mb/s [31], on par with cloud-based game streaming bitrates [38].

This paper presents an analysis of the network congestion response for three commercial cloud-based game streaming systems – Google Stadia, NVidia GeForce Now and Amazon Luna – providing a direct comparison of their bitrates over time and impact on network congestion when competing for scarce capacity with DASH flows over a range of network conditions. We configure and host a DASH server and client in our testbed. Our methodology launches the client and streams the video automatically while running the game systems via a script playing the same game on each system to ensure similar player actions across runs. By necessity, the commercial cloud-based game streaming servers are on the Internet, so as to be as comparable as possible we: 1) interlace runs of each game system serially to minimize temporal differences, and 2) doing 15 runs for each test condition to provide for a large sample.

The results show the three game systems do not have self-induced congestion when there are no competing flows, but do suffer from congestion when competing with DASH flows. How fairly bottleneck capacity is shared depends primarily upon the game system and bottleneck queue sizes, with small bottleneck queues favoring the Stadia and Luna systems, while GeForce is the most fair but less so with a typical bottleneck queue. The frame rates for the cloud-based game systems are always good, but round-trip times can degrade the player quality of experience when bottleneck queues are large. The quality of the DASH streams are best when the bottleneck capacity is shared equally, which primarily only happens for a large bottleneck queue with high capacity. Interruptions to the DASH flow per minute, however, only severely degrades when the bottleneck capacity is low.

The rest of this paper is organized as follows: Section 2 provides related work on measurements of commercial cloud-based game streaming systems and DASH video; Section 3 describes our methodology, including testbed setup and experiment design and parameters; Section 4 analyzes the experimental results; Section 5 discusses the implication of the results; Section 6 mentions limitations and future work; and Section 7 summarizes our conclusions.

[1]There are 123.6 million households in the U.S.

2 RELATED WORK

This section describes work that is related to ours in two main areas: 1) measurements of cloud-based game streaming systems (Section 2.1) and Quality of Experience (QoE), and 2) performance of Dynamic Adaptive Streaming over HTTP (Section 2.2).

2.1 Cloud-based Game Streaming

There are studies analyzing the network performance of early commercial cloud-based game systems, such as OnLive [37] and Gaikai [36]. Manzano et al. [22] collect and analyze network traffic traces from five different games on both OnLive and Gaikai. They find cloud-based game streaming systems have higher bitrates than do traditional network games. Claypool et al. [10] make more detailed analysis and observations of OnLive network traffic traces and find OnLive has network turbulence more akin to high-definition, live video, with large, frequent packets and high bitrates.

For current systems, Suznjevic et al. [32] measure network traffic for NVidia GeForce Now and find GeForce requires bitrates significantly higher than earlier systems (about 25 Mb/s compared to 6 Mb/s previously). Marc et al. [7] limit link capacities for Google Stadia during gameplay, finding Stadia adjusts the resolution and/or frame rates in response to a bitrate reduction. Xu and Claypool [38] measure Google Stadia game traffic for several games, showing Stadia has a traffic pattern similar to but still significantly different than streaming video and at much higher rates than previous cloud-based game systems or video (about 19 Mb/s compared to 6 Mb/s). An extension of this work [40] measures the responses of three commercial systems, finding the three systems have different adaptations to network congestion and vary in their fairness to competing TCP flows sharing a bottleneck link.

While the above papers are helpful for characterizing network characteristics for cloud-based game streaming systems, they do not measure system congestion response when faced with competing DASH flows.

2.2 Dynamic Adaptive Streaming over HTTP

There have been numerous works assessing the Quality of Experience of DASH video. Seufert et al. [30] and Garcia et al. [14] provide surveys of different techniques.

Duanmu et al. [12] compared several QoE models by first conducting a user study to develop a new model and then doing comparisons via simulation. Pastrana et al. [27], Qi et al. [28] and Moorthy et al. [23] measure the impact of interrupts on streaming video and show QoE is impacted by the duration and the frequency of the interrupt events. They find users tend to prefer videos with fewer, even if longer, interrupts.

Hoßfeld et al. [19] and Sackl et al. [29] find fundamental differences between initial delays and interrupts, where initial delays are generally more tolerated than interrupts in the middle of the video stream. These results are confirmed by Allard et al. [4].

Garcia et al. [13] investigate the quality impact of the combined effect of initial loading, interrupts, and compression for high definition sequences, from which they observe an additive impact of interrupts and compression on perceived QoE.

Based on these subjective user studies, the video quality and number of interrupts in the video playout are key factors in our assessment of DASH QoE.

There are numerous evaluations of DASH, as well. A core aspect of DASH performance is the bitrate adaptation algorithm deployed. Bentaleb et al. [5] provide a survey of bitrate adaptation techniques. Bhat et al. [6] evaluate DASH using QUIC versus DASH using TCP with different quality adaptation algorithms. They assess 2v2 flows competing on a measurement testbed, and find QUIC does not provide an immediate benefit to performance versus TCP. Abdelsalam et al. [2] evaluate DASH bitrate adaptation policies over "challenging" network links – those with highly variable capacities. They find adapting bitrates based on throughput leads to frequent resolution changes, whereas adapting bitrates based on buffer occupancy leads to a high number of streaming interruptions. Our work is complementary in that while our focus is on the cloud-based game streaming system, we evaluate the quality of a reference DASH implementation when it competes with a game stream.

3 METHODOLOGY

To observe the response of cloud-based game streaming systems to competing DASH flows, we selected three popular commercial systems and a game common to all (Section 3.1), configured a client and server for DASH streaming (Section 3.2), setup a measurement testbed that allowed for controlling congestion conditions (Section 3.4), gathered network traces (Section 3.5), and analyzed the data (Section 4).

3.1 System and Game Selection

We selected three cloud-based game streaming platforms – Google Stadia, NVidia GeForce Now, and Amazon Luna – based on their current popularity for game players. While Luna and GeForce offer native applications for client-side players, since all three support play via the Google Chrome browser, we use Chrome as the game client for a fair comparison across systems. We also considered Sony Playstation Now and Microsoft XCloud cloud-based game streaming systems, but Playstation Now does not support play through a Web browser (a special App is required) and our preliminary measurements show XCloud appears to have a much lower target quality and bitrate making for unequal comparisons with other systems.

For game selection, as for the platform, we sought a game that could be played on each system to allow for a fair comparison. We selected one of the few games available on all: *Ys VIII: Lacrimosa of Dana* (Nihon Falcom, 2016) – a third person action/exploration game. In our experiments, each Ys run, the game loads the same map and during gameplay, three characters (one controlled by the player) fight enemies for 10 minutes.

Since gameplay visuals (i.e., what is streamed to the client and the player sees) depend upon the player's actions, we wrote scripts to play the game automatically, thus providing identical, repeatable gameplay conditions across runs and across platforms. Our scripts open the game (with input appropriate for each system), load the same game map, and then play the game automatically as might a human player. The script executes player actions, including jump, run, attack, cast abilities and camera rotation, at a frequency and pattern that a human player does (although not necessarily in response to what is happening on the screen). This means the same actions can be repeated exactly across all runs.

3.2 DASH Configuration

The server runs Apache on Linux, hosting a manifest and segments for the DASH configuration of the video Big Buck Bunny[2] encoded into 5 different quality levels for adaptive bitrate scaling. The encoding levels, resolutions and bitrates are shown in Table 1. The DASH client is DASH.js[3] – a reference client implementation for playback of DASH via JavaScript – running on Firefox. The reference client uses the DYNAMIC [31] Adaptive Bitrate Streaming (ABR) algorithm by default.

Table 1: DASH quality levels.

Quality Level	Resolution (pixels)	Bitrate (Mb/s)
1	480x270	2.0
2	640x360	3.0
3	960x540	5.0
4	1280x720	10.0
5	1920x1080	17.5

[2]https://en.wikipedia.org/wiki/Big_Buck_Bunny
[3]https://github.com/Dash-Industry-Forum/dash.js/

3.3 Network Conditions

Our goal is to assess the congestion response for the cloud-based game streaming systems considering congestion arising from both network capacity limits and competing DASH traffic. The network capacity limits alone allow comparison of system responses to possible self-induced congestion arising from various "last-mile" network conditions provided, say, by an Internet Service Provider (ISP), as well as provide baseline performance for constrained conditions without competing DASH flows. Adding competing traffic allows comparison of system congestion responses for co-induced congestion caused by the presence of other network flows on the bottleneck link. We consider link capacities that are: 1) above the maximum required for each system, but that are less than twice the needed capacity when competing with another flow, 2) right at the maximum capacity required, and 3) less than half (40%) of the maximum capacity required.

The dynamics of TCP congestion control algorithms (used by the DASH streams) are influenced by the size of the queue at the bottleneck router. A general rule of thumb is that the bottleneck link's buffer queue size should be a multiple (typically 1x) of the product of the bottleneck capacity and the round-trip delay, otherwise known as the bandwidth-delay product (BDP) – i.e., the BDP is computed by taking the link capacity (bottleneck) in bits per second and multiplying it by the round-trip time (delay) in seconds. Other guidelines suggest a "good" queue size is $(BDP)/\sqrt{n}$, where n is the number of flows at the bottleneck link. However, there are also routers that have considerably larger buffers, a phenomena known as "buffer bloat" [15]. We consider a range of queue sizes, including those that are: 1) shallow, at about one-half the BDP, 2) typical, at about 1x the BDP, and 3) bloated, at about 7x the BDP.

3.4 Measurement Testbed

Figure 1 depicts the general setup for our measurement testbed. Our testbed automatically plays the game YS via Chrome on the game client depicted in the figure, connecting through our custom router to the appropriate cloud-service provider (one of Google Stadia, NVidia GeForce Now or Amazon Luna). For experiments with competing traffic, the bottleneck link (from the router down to the clients) is shared by a DASH client. The game client is a PC running Windows 10 Pro, connecting to the cloud-based game streaming service via Chrome version 98.0.4758.102 (64-bit). The PC hardware is an Intel i7 eight-core CPU @ 2.0 GHz with 64 GB RAM with a 1 Gb/s Ethernet NIC. The PC has an LED monitor with 1920x1080 pixels running at 60 Hz. The DASH client and server are both Alienware PCs each with an 4-core Intel i7-4790K CPU @ 4 GHz with 16 GB RAM running Ubuntu 20.04 LTS, Linux kernel version 5.4, and connect with 1 Gb/s Ethernet NICs.

The game client and DASH client PCs connect via a 1 Gb/s switch to a Raspberry Pi 4 configured to act as a network router. The Pi has a 5 GHz 64-bit quad-core CPU with 8 GB of RAM and runs Ubuntu 20.04 LTS, Linux kernel version 5.4, using `tc` [35] and `netem` to constrain the network capacity and add delay. Wireshark is used to gather all network traces for throughput analysis, gathering the game streaming traffic on the router and the DASH traffic on the DASH client.

Examples of a tc-netem commands run on our router are:

```
tc qdisc add dev eth0 root handle 1: \
  netem delay 4ms
tc qdisc add dev eth0 parent 1: handle 2: \
  tbf rate 15mbit burst 1mbit limit 510kbit
```

The first command adds delay to the system (used to make sure all systems have the same round-trip times) and the second command sets the capacity limit and buffer size.

The router connects to the Internet via our campus network. As a baseline measure of throughput, Google's `M-Lab` Internet speed test consistently shows the campus network through the router to our client PC has downstream bitrates over 900 Mb/s and upstream bitrates over 200 Mb/s. These rates are well-beyond what the streaming services require – i.e., our campus network is not the bottleneck. According to the IP addresses observed and the server location information released by the platforms, the game servers used in our experiments are all on the U.S. east coast, physically near our university.

Based on `ping` measurements from our client, the Stadia servers have an average round-trip time of 11.5 ms, GeForce servers 4.0 ms and Luna servers 16.0 ms. For equal comparison across systems, our router adds 4.5 ms round-trip delay to Stadia, 12.0 ms to GeForce and 15.0 ms for the DASH client to provide about a 16.0 ms round-trip time for all. While a 16.0 ms may be a lower round-trip time than that of many residential connections, our focus is on a comparison of congestion response, not necessarily the quality of the individual connections. The delay is added symmetrically (equal amounts on the up and down streams), while many residential networks may have asymmetric delays. However, since the our delay measurements are primarily used as in indication of Quality of Experience (QOE) and QoE for cloud-based games is affected by round-trip times, the asymmetry should not matter.

3.5 Experiments

Our pilot studies determined 3 minutes of gameplay provided for a steady state bitrate for cloud-based game streams. Our Wireshark traces begin after the game is being played (i.e., loading, menus, etc. are *not* included — just gameplay).

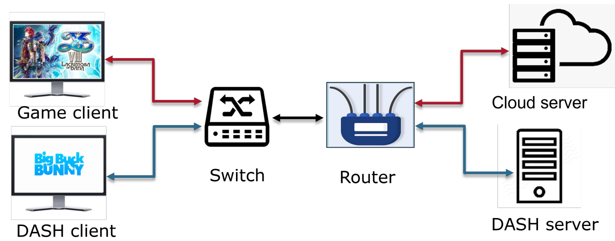


Figure 1: Measurement testbed.

We measured the steady-state bitrates for our systems (Stadia, GeForce and Luna) and selected game (Ys) on an unconstrained network, and the averages are shown in Table 2 with the standard deviation in the parenthesis. All three game systems have the same quality settings – 1080P resolution and 60 f/s framerate. Based on these rates, we tested 3 network conditions: a "good" connection with a capacity limit of 35 Mb/s which is above the baseline bitrates, a "normal" connection with a capacity limit of 25 Mb/s which is right at the baseline bitrates, and a "bad" connection with a capacity limit of 15 Mb/s which is below the baseline bitrates.

Table 2: Game system bitrates without capacity constraints or competing traffic. Units are Mb/s. Mean values are reported with standard deviations in parentheses.

System	Bitrate (Mb/s)
Stadia	28.5 (2.3)
GeForce	27.5 (4.8)
Luna	23.7 (0.9)

Additional scripts automatically: 1) connect to the router to: a) set the queue size and bottleneck capacity limit, as appropriate, and b) launch Wireshark; 2) launch a `ping` command from the client to the game server[4]; and 3) start `presentmon`[5] to record frame rates at the client.

In summary, for each round, the fully-automated experiment procedure is:

(1) Connect to the DASH server.
(2) Start the game in the browser and wait for the game to load.
(3) Connect to the router to set the bitrate, delay and queue size and start Wireshark.
(4) Initiate a `ping` from the client to the appropriate game server and start `presentmon`.
(5) Run the script on the game client which launches and then plays the game Ys.
(6) After 3 minutes, start to play the video streaming to the DASH client.
(7) Continue the script which plays the game Ys for 3 more minutes, then stop the DASH video streaming.
(8) Continue the script which plays the game Ys for a 3 final minutes.
(9) Close the game and all data collection tools and reset the router to the unconstrained conditions.
(10) Repeat the above procedure for each of the three systems (Stadia, GeForce and Luna).

We repeat the above procedure 15 times for each network condition (capacity constraint and router queue size combination), and cloud-based game streaming system. Since Internet conditions from the campus network to the game servers can change over time, we stripe across game service to keep system comparisons as temporally close as possible. For consistency, all this is done by the scripts automatically, without manual intervention. Thus, the order of experimental runs through the parameters from outer loop to inner loop is: [1 to 15 iterations] [B35, B25, B15 capacity constraint] [7x, 1x, 0.5x router queue size] [Stadia, GeForce, Luna game system].

A complete run of all systems and all iterations takes about 24 hours providing for performance that accounts for any time-of-day affects. All runs were completed on a weekday during November 2022.

Table 3 provides a summary of the key experimental parameters.

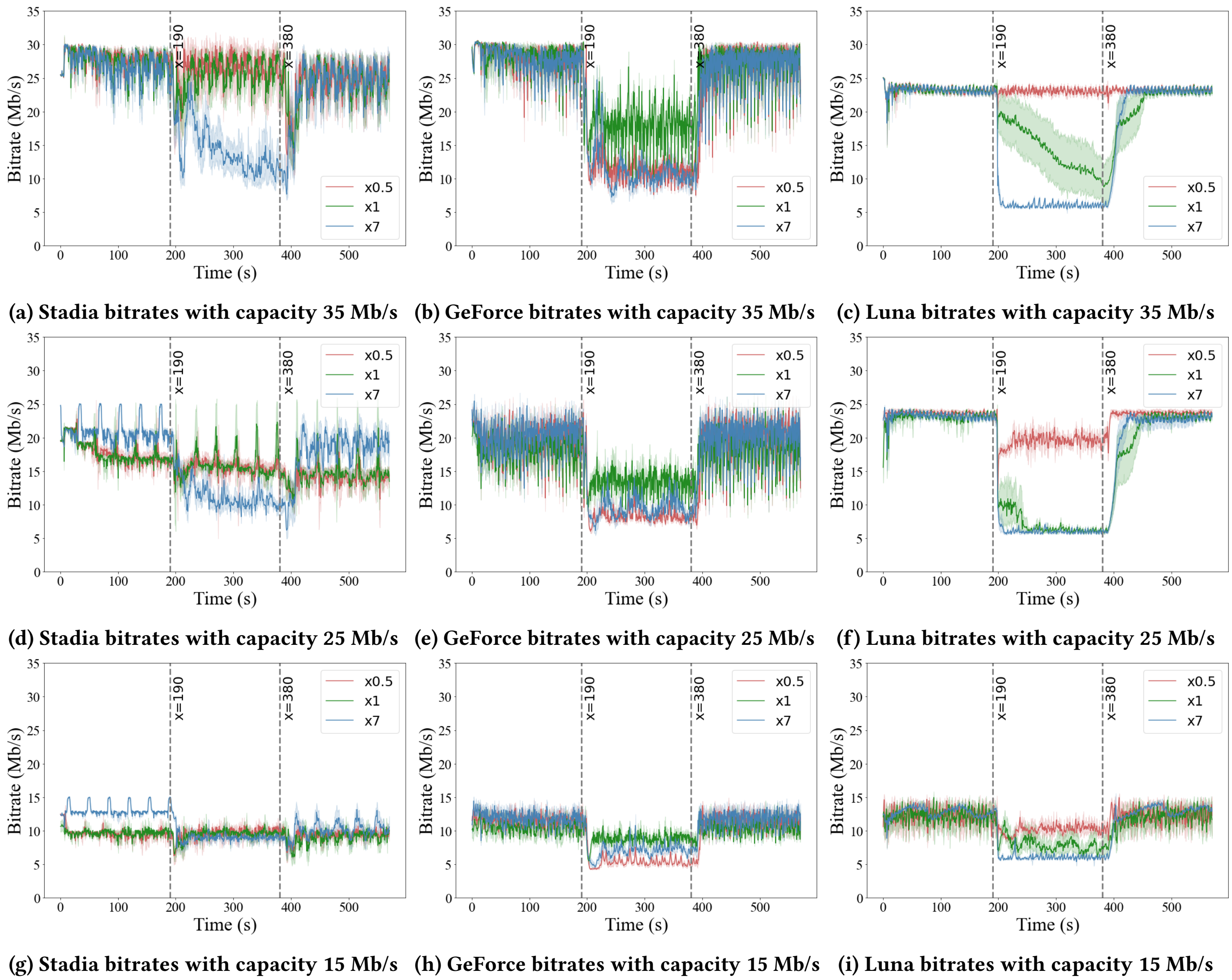
Table 3: Experimental parameters.

Game system	Stadia, GeForce, or Luna
Game	Ys VIII: Lacrimosa of Dana
Capacity limit	15, 25, or 35 Mb/s
Queue size	0.5x, 1x, or 7x BDP
Competing connection	DASH video streaming
Trace length	9 minutes (3 with DASH)
Iterations	15 runs per condition

4 ANALYSIS

This section compares the different cloud-based game streaming systems with capacity and queue limits, both with and without competing DASH flows, considering: 1) game streaming bitrates and DASH video streaming bitrates (Section 4.1); 2) bitrate fairness (Section 4.2); and 3) indicators of quality of experience both for the cloud-based game and the DASH video (Section 4.3).

[4]Identified automatically in a script via the Wireshark trace.
[5]https://github.com/GameTechDev/PresentMon



(a) Stadia bitrates with capacity 35 Mb/s **(b) GeForce bitrates with capacity 35 Mb/s** **(c) Luna bitrates with capacity 35 Mb/s**

(d) Stadia bitrates with capacity 25 Mb/s **(e) GeForce bitrates with capacity 25 Mb/s** **(f) Luna bitrates with capacity 25 Mb/s**

(g) Stadia bitrates with capacity 15 Mb/s **(h) GeForce bitrates with capacity 15 Mb/s** **(i) Luna bitrates with capacity 15 Mb/s**

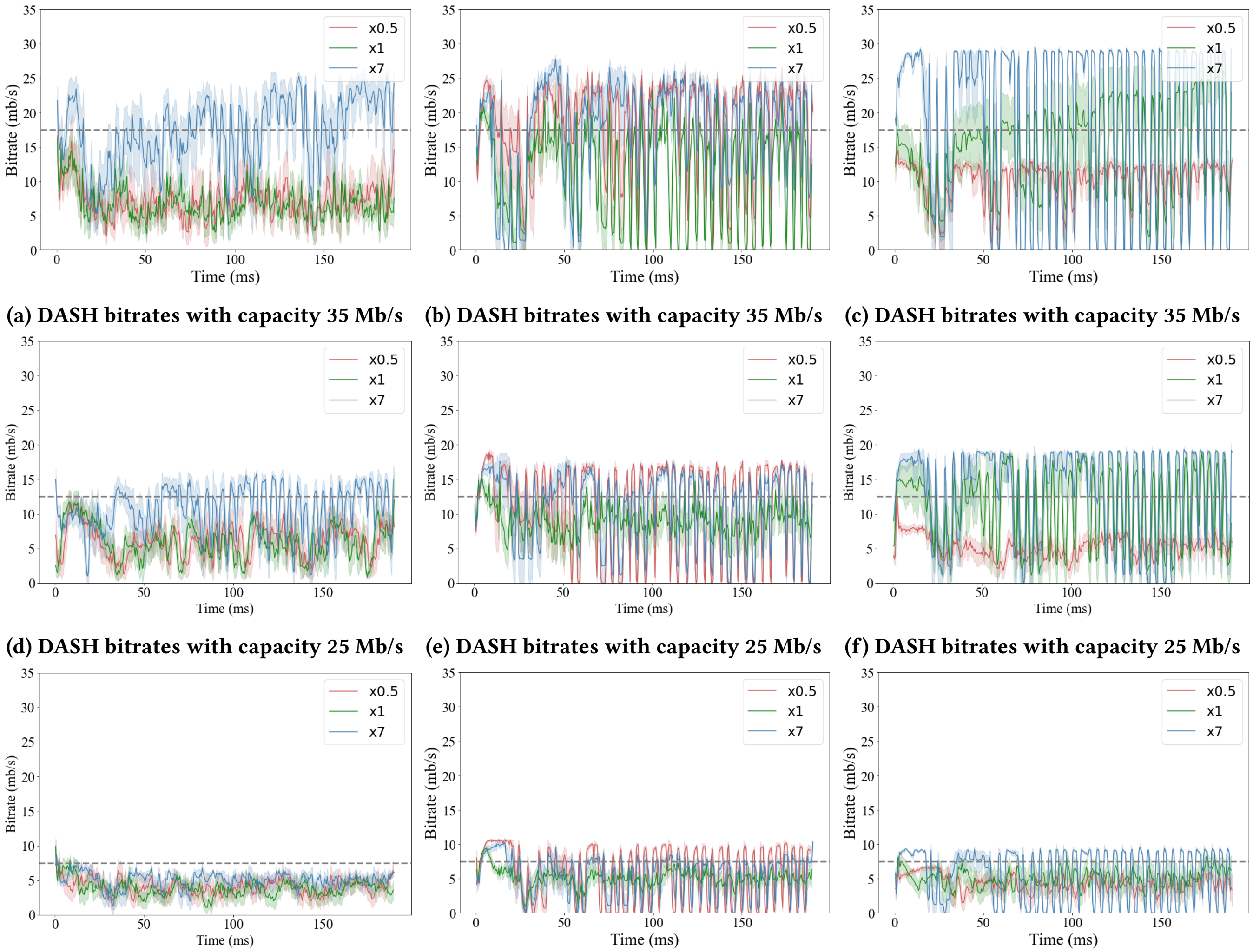
Figure 2: Game system bitrate versus time with a simultaneous DASH flow from 190s to 380s. Each line shows a separate run with a different bottleneck queue size (0.5x, 1x, or 7x) in multiples of the bandwidth delay product (BDP).

4.1 Bitrates

We start analysis with a bitrate comparison (computed every 0.5 seconds) of each cloud-based game streaming system for each queue size (0.5x, 1x and 7x BDP) where the competing DASH flow runs for 3 minutes in the middle of the 9 minute game run. Figure 2 depicts the results for all capacity constraints (35, 25 and 15 Mb/s). For each graph, the x-axis is gameplay time, in seconds, and the y-axis is the measured bitrate, in Mb/s. The mean bitrate for the game system is shown with a colored line with the shading depicting 95% confidence intervals across the 15 runs. There is one line for each queue size: 0.5x BDP - red, 1x BDP - green, and 7x BDP - blue. The left vertical dotted line at 190s shows when the DASH flow starts and the right vertical dotted line at 390s shows when the DASH flow stops.

Before the competing DASH flow arrives (i.e., up to time 190s), mostly the three systems have similar maximum bitrates near the capacity limit except for Luna when the capacity is 35 Mb/s because the maximum bitrate of Luna is around 25 Mb/s. Generally Luna has the least bitrate variation and GeForce and Stadia the most.

When the DASH flows arrive (time 190s), the bitrates for all three systems decrease, indicating they respond to the presence of other traffic competing for the available capacity



(a) DASH bitrates with capacity 35 Mb/s (b) DASH bitrates with capacity 35 Mb/s (c) DASH bitrates with capacity 35 Mb/s

(d) DASH bitrates with capacity 25 Mb/s (e) DASH bitrates with capacity 25 Mb/s (f) DASH bitrates with capacity 25 Mb/s

(g) DASH bitrates with capacity 15 Mb/s (h) DASH bitrates with capacity 15 Mb/s (i) DASH bitrates with capacity 15 Mb/s

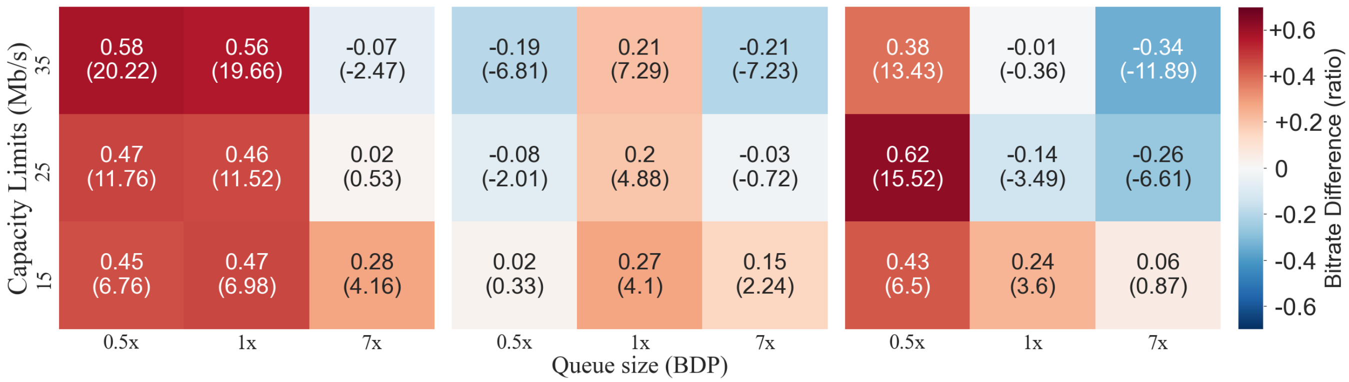
Figure 3: DASH bitrate versus time with a different cloud-base game streaming systems.

and, similarly, the systems recover to their original bitrates sometime after the DASH flows leave (after time 370s). When competing with DASH flows, the Stadia and Luna bitrates depend upon the bottleneck queue sizes, with a larger queue (7x BDP) resulting in a lower bitrate by the game system than with a small queue (0.5x BDP). GeForce bitrates are always highest with the typical queue (1x BDP) and lowest with the small queue (0.5x BDP) and the larger queue (7x BDP).

Figure 3 shows similar bitrate data but for the DASH video streams, which run from 190s to 380s but are shifted to start at time 0 in these graphs. For all capacity constraints, the DASH bitrates fluctuate over time significantly, with the largest fluctuations for the 7x BDP queue, although the 7x queues also generally yield the highest DASH bitrates. As implied by Figure 2, the DASH flows are mostly under their fair-share in capacity, but periodically rise above the fair share for most configurations save for when competing with Stadia at 0.5x and 1x BDP queues.

4.2 Fairness

We next analyze bitrate fairness measured as the difference in bitrates between each game system and the competing DASH flow from time 220s to 370s, normalized by the capacity. This provides fairness measures that range from -1 to +1, with positive values indicating the game system receives a higher portion of the bottleneck capacity and negative numbers indicating the DASH flow receives a higher portion of the bottleneck capacity.



(a) Stadia (b) Geforce (c) Luna

Figure 4: Ratio of bitrate difference (i.e., difference ÷ capacity) for a game system competing with DASH flow.

Figure 4 uses heatmaps to depict the results. The top row of heatmaps is for the game systems competing with DASH flows. There is one large box for each game system (Stadia on the left, GeForce in the middle, and Luna on the right), with the smaller boxes within each representing one network condition – 35, 25, and 15 Mb/s capacities as rows and 0.5x, 1x and 7x BDP queues as columns. The numbers in the boxes are the average difference in throughput for the game system minus the competing TCP flow, shown normalized by the capacity (thus ranging from -1 to +1). The warm, red tones show where the game system has a higher bitrate than the DASH flow and the cool, blue tones where the game system has a lower bitrate.

Visually, Stadia is mostly "warm" and gets more than its fair share of the capacity. In contrast, GeForce has more "cool" areas where it gets slightly less than the fair share, with GeForce having one "warm" column for the typical queue (1x BDP). Luna has half "warm" areas for the small queue (0.5x and 1x BDP) and half "cool" areas for the larger queues (1x and 7x BDP). Stadia is mostly "hot" but does have two "cool" areas with slightly less than the fair share for large queues (7x BDP) and capacity 25 and 35 Mb/s.

4.3 Indicators of Quality of Experience

This section analyzes the indicator of quality of experience (QoE) for both the game flow and the DASH flow.

4.3.1 Game Flow QoE. While formal, game-independent predictors of player QoE in cloud-based games have not been established nor agreed upon, indicators of QoE are delay (which degrades responsiveness) and frame rate (which relates to visual smoothness).

Games played over a cloud-based game streaming system are sensitive to delay since all player input needs to be sent to the server, acted upon by the game engine, rendered, and then sent back to the client before the player can see the outcome of their actions. There are inherent delays in the end systems – e.g., input delay from the mouse and monitor on the client, game engine updates and rendering on the server – but the network round-trip time is "extra" delay that would not be present if the game was played entirely locally. Table 4 (without a competing DASH flow) and Table 5 (with a competing DASH flow) have the round-trip times for the 3 systems for each condition (capacity, queue size). Each value is the mean for 3 minutes of gameplay with standard deviations shown in parentheses. For all systems, when there is no competing DASH flow, the round-trip times are low,

Table 4: Round-trip time (ms) without competing DASH flow.

	BDP 0.5x			BDP 1x			BDP 7x		
Capacity	Stadia	GeForce	Luna	Stadia	GeForce	Luna	Stadia	GeForce	Luna
15 Mb/s	16.1 (1.7)	16.2 (1.1)	16.8 (2.2)	17.3 (4.3)	16.5 (2.4)	18.4 (5.2)	21.5 (14.0)	22.0 (18.6)	16.5 (5.3)
25 Mb/s	16.4 (2.2)	16.2 (1.1)	16.0 (0.9)	17.2 (4.3)	16.5 (2.3)	16.0 (1.1)	24.8 (6.0)	18.4 (8.7)	15.9 (0.6)
35 Mb/s	15.7 (0.6)	16.0 (0.5)	16.0 (0.9)	15.6 (0.7)	16.0 (0.4)	16.0 (0.8)	15.7 (0.6)	16.0 (0.5)	16.0 (0.7)

Table 5: Round-trip time (ms) with competing DASH flow.

	BDP 0.5x			BDP 1x			BDP 7x		
Capacity	Stadia	GeForce	Luna	Stadia	GeForce	Luna	Stadia	GeForce	Luna
15 Mb/s	17.5 (2.6)	16.6 (1.6)	17.2 (2.3)	22.4 (5.4)	21.9 (5.4)	22.4 (5.4)	89.0 (39.9)	67.4 (41.8)	68.2 (46.9)
25 Mb/s	17.5 (2.6)	16.7 (1.6)	17.7 (2.6)	22.2 (5.9)	21.8 (5.3)	21.6 (5.8)	80.4 (41.0)	72.1 (40.3)	76.6 (45.0)
35 Mb/s	17.5 (2.6)	17.0 (1.8)	17.1 (2.2)	22.2 (5.5)	21.1 (5.4)	23.3 (5.7)	83.0 (39.3)	80.6 (40.1)	74.8 (47.3)

Table 6: Game frame rate (f/s) without competing DASH flow.

	BDP 0.5x			BDP 1x			BDP 7x		
Capacity	Stadia	GeForce	Luna	Stadia	GeForce	Luna	Stadia	GeForce	Luna
15 Mb/s	59.2 (0.1)	58.4 (0.5)	53.8 (1.9)	58.9 (0.3)	58.0 (0.7)	52.2 (2.9)	59.7 (0.1)	58.5 (0.4)	59.8 (0.1)
25 Mb/s	58.7 (0.3)	57.5 (0.9)	59.9 (0.2)	58.5 (0.2)	58.0 (0.7)	59.2 (0.6)	59.7 (0.1)	59.5 (0.1)	59.9 (0.1)
35 Mb/s	59.8 (0.0)	59.9 (0.1)	59.9 (0.1)	59.8 (0.0)	59.9 (0.1)	59.9 (0.1)	59.8 (0.1)	59.9 (0.1)	59.9 (0.1)

Table 7: Game frame rate (f/s) with competing DASH flow.

	BDP 0.5x			BDP 1x			BDP 7x		
Capacity	Stadia	GeForce	Luna	Stadia	GeForce	Luna	Stadia	GeForce	Luna
15 Mb/s	53.1 (0.6)	39.8 (12.7)	55.1 (0.7)	55.4 (0.4)	55.4 (7.0)	56.2 (2.0)	56.0 (0.6)	46.7 (12.1)	57.3 (1.0)
25 Mb/s	55.7 (0.9)	57.6 (0.6)	53.5 (0.6)	57.5 (0.7)	57.1 (1.0)	59.1 (0.5)	57.3 (0.3)	55.5 (4.4)	58.3 (0.2)
35 Mb/s	54.6 (0.8)	57.1 (0.8)	57.8 (1.0)	56.6 (1.0)	57.7 (0.6)	58.9 (0.4)	57.1 (0.3)	55.2 (5.7))	58.4 (0.2)

near minimal (about 16 ms) for small queues, but increasing by about 50% for Stadia and GeForce for larger queues and small capacities. Luna always has lower round-trip times possibly since it has slightly lower maximum bitrates. The small differences in delays between systems may be noticeable to users, but are small enough to not appreciably affect performance or QoE [1]. These round-trip times do not reach even the delays that would be caused by a full queue suggesting that the systems themselves do not saturate a link's available capacity until there is loss.

When there is a competing DASH flow, the round-trip times are still less than the limit dictated by the queue size. In other words, the 0.5 BDP queue can have at most an extra 8 ms of delay from queueing, the 1x queue 16 ms and the 7x queue 112 ms but average values mostly fall short of these numbers. The 0.5x and 1x BDP queues both provide for a good QoE for the game players. The 7x queues appear only about $\frac{1}{2}$ to $\frac{2}{3}$ full on average, slightly higher for Stadia than GeForce and Luna, but add a moderate amount of delay to the game stream degrading QoE. This illustrates how large router queues in the presence of competing flows during congestion result in added delay for game-streaming systems. This delay has been shown to degrade player performance and quality of experience [8]

Frame rate is another key indicator of the game quality, where higher frame rates can improve player performance and are generally associated with a better player quality of experience [9, 20, 21, 42].

Table 6 (without a competing DASH flow) and Table 7 (with a competing DASH flow) have the frame rates for the 3 systems for each condition (capacity, queue size). Each value is the mean for the 3 minutes of gameplay with standard deviations shown in parentheses. Without a competing DASH flow, the frame rates of all systems for all network conditions are near the maximum 60 f/s. When competing with a DASH flow, frame rates are generally a little bit lower compared with frame rates when there is no competing DASH flow, but are still high. For higher link capacities (35 and 25 Mb/s), frame rates are all high (greater than 55 f/s), but for lower link capacity (15 Mb/s), frame rates for all systems decline, especially for GeForce with 0.5x and 7x BDP queues. At 15 Mb/s capacity, GeForce only gets high frame rates with typical queues (1x BDP).

4.3.2 DASH Flow QoE. While there are several models for DASH QoE that could be used, since a single metric has not been agreed upon, we use video throughput, interrupts (stalls) and video quality switch rates as indicators of the QoE for a DASH video [33].

Our DASH video is encoded with 5 quality levels (see Table 1) and with higher bitrates, the video can stream at higher resolutions. Table 8 shows the DASH bitrate when competing with the different cloud-based game systems. Each value is the mean for the 3 minutes the video played with standard deviations shown in parentheses. Generally for small and typical queues (0.5x and 1x BDP), average bitrates are less than half of the link capacities when competing with all game systems, except for GeForce at 35 Mb/s capacity. With large queues (7X BDP), DASH flows generally get more of the link capacities, particularly for high capacities (25 and 35 Mb/s), and, hence have higher frame resolutions. When competing with Stadia, DASH flows get less of the capacity and have worse frame resolutions.

When the client playout buffer runs out, the video playout is interrupted and the buffer is refilled. Users generally find these interrupts annoying [4] – the interrupt rate provides another indication of the QoE for a DASH video. Table 9 shows the number of interrupts per minute for the DASH stream when competing with the different cloud-based game systems. The interrupt rates are low in most network conditions and for most game systems – well under once a minute. Only for DASH flows competing with Stadia at restricted (15 Mb/s) capacity do interrupts average about once every 2 minutes. At this same capacity restriction, GeForce and Luna have a moderate interrupt rate. Overall, DASH streams competing with GeForce have the lowest interrupt rates.

With DASH, the video quality resolutions adapt to measures of throughput and playout buffer occupancy. Generally, users prefer a steady, unchanging visual quality over variable visual quality [14, 30]. Table 10 shows the number of quality

Table 8: DASH bitrate (Mb/s).

	BDP 0.5x			BDP 1x			BDP 7x		
Capacity	Stadia	GeForce	Luna	Stadia	GeForce	Luna	Stadia	GeForce	Luna
15 Mb/s	3.9 (0.4)	5.1 (0.1)	4.1 (0.8)	3.6 (0.2)	4.9 (0.3)	4.9 (0.5)	4.9 (0.4)	5.0 (0.2)	5.2 (0.1)
25 Mb/s	5.8 (1.1)	10.4 (0.1)	4.9 (0.6)	5.5 (0.8)	8.7 (1.3)	10.3 (0.5)	10.5 (0.3)	10.6 (0.8)	12.6 (0.8)
35 Mb/s	6.5 (2.3)	18.2 (0.4)	9.7 (1.1)	6.4 (1.9)	10.9 (0.5)	14.4 (3.8)	16.6 (2.1)	18.2 (1.0)	17.9 (0.4)

Table 9: DASH interrupts per minute.

	BDP 0.5x			BDP 1x			BDP 7x		
Capacity	Stadia	GeForce	Luna	Stadia	GeForce	Luna	Stadia	GeForce	Luna
15 Mb/s	0.49 (0.27)	0 (0)	0.24 (0.15)	0.51 (0.29)	0.16 (0.17)	0.49 (0.34)	0.16 (0.17)	0.02 (0.08)	0 (0)
25 Mb/s	0.20 (0.20)	0 (0)	0.07 (0.13)	0.11 (0.16)	0 (0)	0.02 (0.08)	0.13 (0.16)	0 (0)	0 (0)
35 Mb/s	0.09 (0.19)	0 (0)	0 (0)	0.11 (0.16)	0 (0)	0 (0)	0.09 (0.15)	0 (0)	0 (0)

Table 10: DASH quality switches per minute.

	BDP 0.5x			BDP 1x			BDP 7x		
Capacity	Stadia	GeForce	Luna	Stadia	GeForce	Luna	Stadia	GeForce	Luna
15 Mb/s	4.3 (1.1)	1.1 (0.3)	3.0 (1.5)	4.3 (1.3)	4.1 (1.5)	3.4 (1.6)	4.1 (1.8)	1.2 (0.8)	1.0 (0.2)
25 Mb/s	4.1 (1.0)	1.8 (0.2)	3.4 (1.6)	4.0 (1.5)	4.2 (1.1)	1.7 (0.9)	4.3 (1.1)	2.0 (0.6)	1.9 (0.4)
35 Mb/s	3.3 (1.4)	2.6 (0.8)	4.0 (1.0)	4.4 (2.0)	4.3 (0.5)	3.0 (1.5)	4.6 (1.4)	3.2 (1.4)	1.6 (0.2)

switches per minute – i.e,. the DASH video switched from one level to another in the encoded video in Table 1 during playout. In general, when competing with Luna and GeForce, the DASH video streams have fewer quality switches per minute than when competing with Stadia, and the quality switches per minute drop substantially for GeForce and Luna for large bottleneck queues (7X BDP). GeForce, has similar rates of quality switching for both 0.5x and 7x BDP queues.

5 DISCUSSION

Previous work [39] examines cloud-based game streaming system response to competing bulk-downloads of iperf flows. The results show competing with bulk-downloads impact game stream QoE considerably, especially for GeForce and Luna. Large bottleneck queues and limited capacities especially degrade QoE, increasing round-trip times 7-fold and halving frame rates. In contrast, competing DASH flows have far less impact. While QoE degrades somewhat with bloated queue capacities that cause higher round-trip times, round-trip times are usually low and frame rates generally high, near 60 f/s for all scenarios. The competing DASH flows do not fair as well, typically getting less than half the capacity and frequently quality switches, although interrupts remain low.

In general, there are significant differences in congestion response across the three systems – i.e., there is no "one size, fits all". This suggests measurement studies should consider more than one system in order to determine representative behavior. The good news is that all of the streaming systems do, in fact, appear to respond to congestion, even if competing DASH flows tend to get less than their fair share of the bottleneck capacities in most cases. However, the ability (and perhaps willingness) of cloud-based game streaming systems to adapt when the capacities are more restricted (15 Mb/s) is limited, and generally small bottleneck queues result in a less adaptive cloud-based game streaming flow.

Indicators of quality are that game streaming flows will provide a good player experience, even in the presence of competing DASH video, which bodes well for the continued growth of cloud-based game streaming. However, more complete measures of QoE should still be sought and could perhaps include measures of video quality using tools such as NDNetGaming [34] or Google UVQ [41].

6 LIMITATIONS AND FUTURE WORK

Our focus is on the view of the game player, where the game session on a cloud-based game streaming system must compete with an arriving streaming video in the middle of gameplay. The converse scenario – a streaming video started first, then later competing with an arriving game session – is also quite likely and may yield different network and quality of experience results, both for the game streams and the video streams. A future study could use the same methodology employed here, just swapping the timing for the game system with the DASH video.

Our experiments are for one game only (Ys VIII) and prior work has shown that the bitrates for different games on the same cloud-based game streaming system can vary considerably [38]. Future work could see if the comparative differences illustrated in our paper hold for other games, as well. Similarly, whether the results hold for other prominent cloud-based game streaming systems such as those by Microsoft or Sony is not known and could be studied.

This paper focuses on competition from only one DASH flow since this is a reasonable starting point and likely experienced by many cloud-based game streams in many households. Future work could consider more complicated network scenarios with multiple DASH flows, or even mixtures of different types of network flows (e.g., DASH and Web browsing and bulk-downloads). Other experiments could consider mobile networks such as 4G and 5G.

Our router uses only a drop-tail queue, whereas Active Queue Management approaches (AQM) that signal congestion earlier (e.g., Flow Queue CoDel [18] or PIE [26]) might yield different results and could be considered for future study.

7 CONCLUSIONS

Emerging cloud-based game streaming systems hold out the promise of providing a convenient gaming experience for players as long as the network conditions are adequate. In particular, streaming computer games need high definition frames sent at high frame rates (typical targets are HD and 60 f/s). This, in turn, requires high bitrates that have the potential to congest last mile residential networks, particularly when competing for capacity with other network flows. This paper compares three commercial systems – Google Stadia, NVidia GeForce Now and Amazon Luna – with repeated runs of the same game on network links with different capacity constraints and bottleneck queue sizes, while the game systems compete for bottleneck capacity with a DASH flow.

Analysis of the results show the three game systems have similar bitrates that operate near the capacity constraints, and even for constrained conditions (lower capacity, smaller queue sizes) none of the three systems has self-induced congestion, keeping packet queuing low and packet loss minimal in the absence of competing traffic. When competing with a DASH flow, GeForce generally shares the available capacity fairly, Stadia dominates taking about twice what is fair, and Luna take more than it's fair share of capacity for small router queues but less than its fair share for large queues and high capacities.

Large bottleneck queues (buffer bloat) result in larger delays for the game systems, which is bad for game player quality of experience. The game stream frame rates are generally good for all conditions tested – even for those where the game stream is competing with a DASH flow – but constrained network conditions also result in slightly lower frame rates.

The competing DASH flows mostly operate without interruptions in their playout, except for capacity constrained conditions, especially when competing with Stadia. The DASH streams switch quality values frequently under most conditions – generally not desirable for viewers – but tend to be most stable when competing with GeForce and for Luna with large bottleneck queues.

These results provide a better understanding of game system interactions with constrained network links when competing with DASH flows and should be useful to better plan for, and hopefully deter, resulting network congestion, thus potentially improving game player and video viewer Quality of Experience.

REFERENCES

[1] Maha Abdallah, Carsten Griwodz, Kuan-Ta Chen, Gwendal Simon, Pin-Chun Wang, and Cheng-Hsin Hsu. 2018. Delay-Sensitive Video Computing in the Cloud: A Survey. *ACM Transactions on Multimedia Computing, Communications, and Applications* 13, 3s (June 2018), 1–29.

[2] A. Abdelsalam, M. Luglio, M. Quadrini, C. Roseti, and F. Zampognaro. 2020. Analysis of DASH Performance over Time-varying End-to-end Links. *Computers and Electrical Engineering* 84 (2020).

[3] Hamed Ahmadi, Sepideh Khoshnood, Mahmoud Reza Hashemi, and Shervin Shirmohammadi. 2013. Efficient Bitrate Reduction using a Game Attention Model in Cloud Gaming. In *Proceedings of the IEEE International Symposium on Haptic Audio Visual Environments and Games (HAVE)*. 103–108.

[4] Josh Allard, Andrew Roskuski, and Mark Claypool. 2020. Measuring and Modeling the Impact of Buffering and Interrupts on Streaming Video Quality of Experience. In *Proceedings of the 18th International Conference on Advances in Mobile Computing & Multimedia (MoMM)*. Chiang Mai, Thailand.

[5] Abdelhak Bentaleb, Bayan Taani, Ali Begen, Christian Timmerer, and Roger Zimmermann. 2019. A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP. *IEEE Communications Surveys & Tutorials* 21, 1 (2019).

[6] D. Bhat, A. Rizk, and M. Zink. 2017. Not so QUIC: A Performance Study of DASH over QUIC. In *Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*. Taipei, Taiwan.

[7] Marc Carrascosa and Boris Bellalta. 2020. Cloud-gaming: Analysis of Google Stadia Traffic. (2020). arXiv:cs.NI/2009.09786

[8] Kuan-Ta Chen, Yu-Chun Chang, Hwai-Jung Hsu, De-Yu Chen, Chun-Ying Huang, and Ching-Hsien Hsu. 2014. On the Quality of Service of Cloud Gaming Systems. *IEEE Transactions on Multimedia* 16, 2 (2014).

[9] Kajal Claypool and Mark Claypool. 2007. On Frame Rate and Player Performance in First Person Shooter Games. *Springer Multimedia Systems Journal (MMSJ)* 13 (2007), 3–17.

[10] Mark Claypool, David Finkel, Alexander Grant, and Michael Solano. 2012. Thin to Win? Network Performance Analysis of the OnLive Thin Client Game System. In *11th Annual Workshop on Network and Systems Support for Games (NetGames)*. 1–6.

[11] Andrea Di Domenico, Gianluca Perna, Martino Trevisan, Luca Vassio, and Danilo Giordano. 2021. A Network Analysis on Cloud Gaming: Stadia, GeForce Now and PSNow. *Network* 1, 3 (2021).

[12] Zhengfang Duanmu, Kai Zeng, Kede Ma, Abdul Rehman, and Zhou Wang. 2017. A Quality-of-Experience Index for Streaming Video. *IEEE Journal of Selected Topics in Signal Processing* 11, 1 (Feb. 2017).

[13] M. Garcia, D. Dytko, and A. Raake. 2014. Quality Impact due to Initial Loading, Stalling, and Video Bitrate in Progressive Download Video Services. In *Proceedings of the International Conference on Quality of Multimedia Experience (QoMEX)*. Singapore.

[14] M. Garcia, F. De Simone, S. Tavakoli, N. Staelens, S. Egger, K. Brunnström, and A. Raake. 2014. Quality Of Experience and HTTP Adaptive Streaming: A Review Of Subjective Studies. In *Proceedings of the IEEE International Conference on Quality of Multimedia Experience (QoMEX)*. Singapore.

[15] Jim Gettys and Kathleen Nichols. 2011. Bufferbloat: Dark Buffers in the Internet: Networks without Effective AQM May Again be Vulnerable to Congestion Collapse. *ACM Queue* 11 (Nov. 2011), 40–54.

[16] Philippe Graff, Xavier Marchal, Thibault Cholez, Stéphane Tuffin, Bertrand Mathieu, and Olivier Festor. 2021. An Analysis of Cloud Gaming Platforms Behavior under Different Network Constraints. In *Proceedings of the 17th International Conference on Network and Service Management (CNSM)*. Izmir, Turkey, 551–557.

[17] IMARC Group. 2022. Cloud Game Market. (Feb. 2022). https://www.imarcgroup.com/cloud-gaming-market [Accessed 24-Feb-2022].

[18] T. Hoeiland-Joergensen, P. McKenney, D. Taht, J. Gettys, and E. Dumazet. 2018. The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm. *IETF Request for Comments (RFC) 8290* (Jan. 2018).

[19] T. Hoßfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen. 2012. Initial Delay vs. Interruptions: Between the Devil and the Deep Blue Sea. In *Proceedings of IEEE QoMEX*.

[20] J. Huhti. 2019. *The Effect of High Monitor Refresh Rate on Game Quality of Experience*. Ph.D. Dissertation. University of Vaasa, Vaasa, Finland.

[21] Joohwan Kim, Josef Spjut, Morgan McGuire, Alexander Majercik, Ben Boudaoud, Rachel Albert, and David Luebke. 2019. Esports Arms Race: Latency And Refresh Rate For Competitive Gaming Tasks. *Journal of Vision* 19, 10 (Sept. 2019).

[22] M. Manzano, J. A. Hernández, M. Urueña, and E. Calle. 2012. An Empirical Study of Cloud Gaming. In *Proceeding of the 11th Annual Workshop on Network and Systems Support for Games (NetGames)*. 1–2.

[23] A. K. Moorthy, L. K. Choi, A. C. Bovik, and G. De Veciana. 2012. Video Quality Assessment on Mobile Devices: Subjective, Behavioral and Objective Studies. *IEEE Journal Selected Topics in Signal Processing* 6, 6 (Oct. 2012), 652––671.

[24] Nielsen. 2022. State of Play. (April 2022). Online: https://www.nielsen.com/insights/2022/state-of-play/.

[25] Nielsen. 2022. Streaming Grew its Audience in 2021. (Jan. 2022). Online: https://www.nielsen.com/insights/2022/streaming-grew-its-audience-in-2021-drama-reality-and-kids-programming-lead-the-content-wars.

[26] R. Pan, P. Natarajan, C. Piglione, M. S. Prabhu, , V. Subramanian, F. Baker, and B. VerSteeg. 2013. PIE: A Lightweight Control Scheme to Address the Bufferbloat Problem. In *Proceedings of the IEEE 14th International Conference on High Performance Switching and Routing (HPSR)*. Taipei, Taiwan.

[27] R. R. Pastrana-Vidal, J. C. Gicquel, C. Colomes, and H. Cherifi. 2004. Sporadic Frame Dropping Impact on Quality perception. In *Proceedings of the Society of Photo-Optical Instrumentation Engineers Conference*.

[28] Y. Qi and M. Dai. 2006. The Effect of Frame Freezing and Frame Skipping on Video Quality. In *Proceedings of the IEEE International Conference on Intelligent Information Hiding and Multimedia*. Pasadena, CA, USA.

[29] A. Sackl, S. Egger, and R. Schatz. 2013. Where's the Music? Comparing the QoE Impact of Temporal Impairments between Music and Video Streaming. In *Proceedings of the 10th International Conference on Quality of Multimedia Experience (QoMEX)*. Klagenfurt am Worthersee, Austria.

[30] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia. 2014. A Survey on Quality of Experience of HTTP Adaptive Streaming. *IEEE Communication Surveys Tutorial* 17, 1 (Sept. 2014), 469 –– 492.

[31] Kevin Spiteri, Ramesh Sitaraman, and Daniel Sparacio. 2019. From Theory to Practice: Improving Bitrate Adaptation in the DASH Reference Player. *ACM Trans. Multimedia Comput. Commun. Appl.* 15, 2s, Article 67 (jul 2019), 29 pages. https://doi.org/10.1145/3336497

[32] Mirko Suznjevic, Ivan Slivar, and Lea Skorin-Kapov. 2016. Analysis and QoE Evaluation of Cloud Gaming Service Adaptation under Different Network Conditions: The Case of NVIDIA GeForce NOW. In *Proceedings of the 8th International Conference on Quality of Multimedia Experience (QoMEX)*. Lisbon, Portugal, 1–6.

[33] Christian Timmerer. 2015. QoE with High Definition Video Through Adaptive Streaming with MPEG-DASH. (May 2015). Online: https://bitmovin.com/ultra-high-definition-quality-experience-mpeg-dash-part-1/.

[34] Markus Utke, Saman Zadtootaghaj, Steven Schmidt, Sebastian Bosse, and Sebastian Möller. 2022. NDNetGaming - development of a no-reference deep CNN for gaming video quality prediction. *Multimedia Tools and Applications* 81 (01 2022). https://doi.org/10.1007/s11042-020-09144-6

[35] Wikipedia contributors. 2020. Tc (Linux). (2020). https://en.wikipedia.org/wiki/Tc_(Linux) [Accessed 27-Jan-2020].

[36] Wikipedia contributors. 2022. Gaikai. (2022). https://en.wikipedia.org/wiki/Gaikai [Accessed 22-Jan-2022].

[37] Wikipedia contributors. 2022. OnLive. (2022). https://en.wikipedia.org/wiki/OnLive [Accessed 15-Jan-2022].

[38] Xiaokun Xu and Mark Claypool. 2021. A First Look at the Network Turbulence for Google Stadia Cloud-based Game Streaming. In *Proceedings of the IEEE Global Internet Symposium (GI)*. Virtual Conference.

[39] Xiaokun Xu and Mark Claypool. 2022. Measurement of Cloud-Based Game Streaming System Response to Competing TCP Cubic or TCP BBR Flows. In *Proceedings of the 22nd ACM Internet Measurement Conference*. Nice, France, 305–316.

[40] Xiaokun Xu and Mark Claypool. 2022. Measurement of the Responses of Cloud-based Game Streaming to Network Congestion. In *Proceedings of the 32nd ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*. Athlone, Ireland.

[41] Yilin. 2022. UVQ: Measuring YouTube's Perceptual Video Quality. (2022). Online: https://ai.googleblog.com/2022/08/uvq-measuring-youtubes-perceptual-video-quality.html.

[42] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hossfeld. 2010. Impact of Frame Rate and Resolution on Objective QoE Metrics. In *Proceedings of the 2nd International Workshop on Quality of Multimedia Experience (QoMEX)*. Trondheim, Norway.