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

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

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

***Abstract*—Cloud-based game streaming requires low-latency and high-throughput networking to support real-time interactivity and high-quality visuals critical to such platforms. However, the capacity of local networks to meet these demands is challenged by the simultaneous presence of other bandwidth-intensive applications, such as Dynamic Adaptive Streaming over HTTP (DASH) for video content. While some aspects of network congestion for cloud-based game streaming have been studied, missing are comparative performance and congestion responses for cloud-based game streams competing with video flows – a common scenario for users on a home Local Area Network (LAN). This paper presents results from experiments that measure how two commercial cloud-based game streaming systems – NVIDIA GeForce Now and Amazon Luna – respond to 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 and interrupts for the DASH flows show markedly different responses to the arrival and departure of competing DASH traffic.**

I. 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, NVidia GeForce Now, and Amazon Luna with Meta's Facebook gaming arriving soon.

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 for streaming video, the cloud-based game player interacts with the stream frequently and the client sends the player's game input back up to the server to be acted upon in the game. This means a significant disadvantages of cloud-based game streaming over traditional games is the increased traffic required – bitrates for frequent player actions and high-quality video frames can cause congestion, degrading player quality of experience especially in the presence of co-located network traffic.

Previous work has compared the congestion response for cloud-based game streaming systems competing with bulk-downloads [1], whereas a common potential congestion 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. Moreover, the bitrate requirements for video streaming services have significantly increased as well, with more support for 4K UHD content and live sports, including e-sports, that can be as high as 35 Mb/s [2], on par with cloud-based game streaming bitrates [3].

This paper presents an analysis of the network congestion response for two commercial cloud-based game streaming systems – 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) do 15 runs for each test condition to provide for a large sample.

The results show both 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 Luna, while GeForce is more fair but less so with a typical bottleneck queue size.

The rest of this paper is organized as follows: Section II provides related work; Section III describes our methodology; Section IV analyzes the experimental results; Section V discusses the implications; Section VI mentions limitations and future work; and Section VII summarizes our conclusions.

979-8-3503-5209-2/24/$31.00

II. RELATED WORK

This section describes related work: 1) measurements of cloud-based game streaming systems and Quality of Experience (QoE), and 2) performance of Dynamic Adaptive Streaming over HTTP.

A. *Cloud-based Game Streaming*

Suznjevic et al. [4] 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 [5]). Marc et al. [6] limit link capacities for Google Stadia during gameplay, finding Stadia adjusts the resolution and/or frame rates in response to a bitrate reduction. An extension of this work [7] 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.

B. *Dynamic Adaptive Streaming over HTTP*

There have been numerous works assessing the Quality of Experience of DASH video, including survey results by Seufert et al. [8] and Garcia et al. [9]. In particular, Garcia et al. [10] 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 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. [11] provide a survey of bitrate adaptation techniques. Bhat et al. [12] evaluate DASH using QUIC versus DASH using TCP with different quality adaptation algorithms. 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.

III. METHODOLOGY

To assess the response of cloud-based game streaming systems to competing DASH flows, we selected two commercial systems and a game common to both (Section III-A), configured a client and server for DASH streaming (Section III-B), setup a measurement testbed that allowed for controlling congestion conditions (Section III-D), gathered network traces (Section III-E), and analyzed the data (Section IV).

A. *System and Game Selection*

We selected two cloud-based game streaming systems – 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 both support play via the Google Chrome browser, we use Chrome as the game client for a fair comparison across 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 both: *Ys VIII: Lacrimosa of Dana* (Nihon Falcom, 2016) – a third person action/exploration game.

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.

B. *DASH Configuration*

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

TABLE I: DASH quality levels.

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

C. *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 within the range of many U.S. residential connections, one of: 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, or 3) less than half (40%) of the maximum capacity required.

[1] https://en.wikipedia.org/wiki/Big_Buck_Bunny

[2] https://github.com/Dash-Industry-Forum/dash.js/

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 small 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" [13]. We consider a range of queue sizes, including those that are one of: 1) shallow, at about one-half the BDP, 2) typical, at about 1x the BDP, or 3) bloated, at about 7x the BDP.

D. 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 (either 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 1.5 GHz 64-bit quad-core CPU with 8 GB of RAM and runs Ubuntu 20.04 LTS, Linux kernel version 5.4, using `tc` [14] 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.

Based on `ping` measurements from our client, round-trip times to GeForce servers average 4 ms and Luna servers average 16 ms. For equal comparison across systems, our router adds 12 ms to GeForce and 15 for the DASH client to provide about a 16 round-trip time for all flows.

E. Experiments

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

We measured the steady-state bitrates for our systems (GeForce and Luna) and selected game (Ys) on an unconstrained network, and the averages are shown in Table II with the standard deviations in parenthesis. Both game systems have the same quality settings – 1080P resolution and a 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.

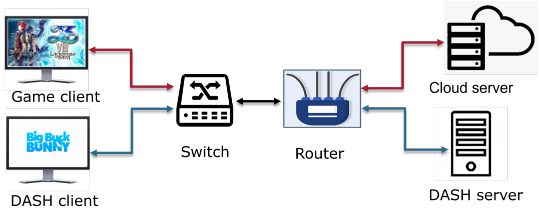


Fig. 1: Measurement testbed.

TABLE II: 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)
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[3]; and 3) start `presentmon`[4] to record frame rates at the client.

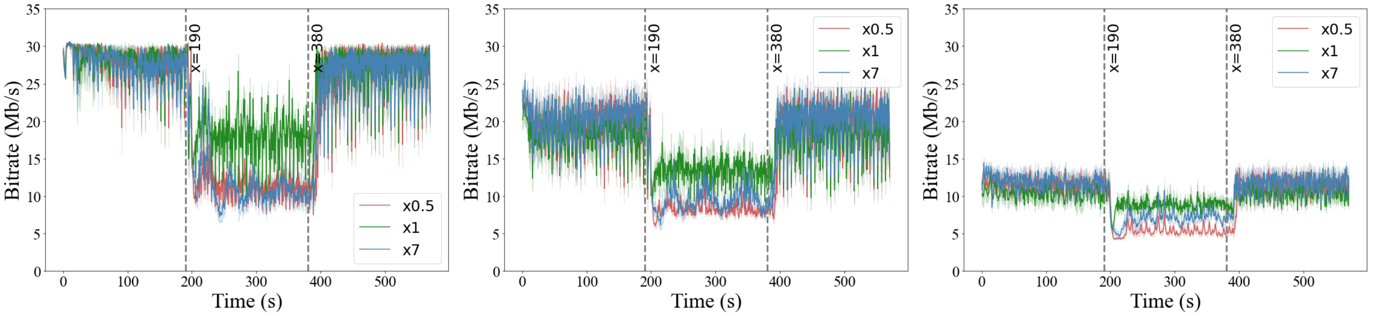
In summary, for each round, the fully-automated experiment procedure is:

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 capacity limit, 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.

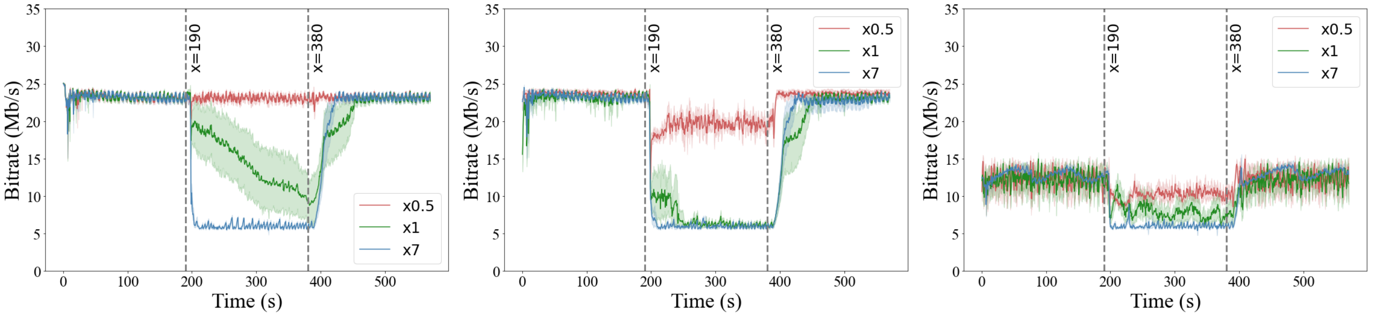
---

[3] Identified automatically in a script via the Wireshark trace.

[4] https://github.com/GameTechDev/PresentMon



(a) GeForce bitrates with capacity 35 Mb/s (b) GeForce bitrates with capacity 25 Mb/s (c) GeForce bitrates with capacity 15 Mb/s



(d) Luna bitrates with capacity 35 Mb/s (e) Luna bitrates with capacity 25 Mb/s (f) Luna bitrates with capacity 15 Mb/s

Fig. 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).

10) Repeat the above procedure for both systems (GeForce and Luna).

We repeat the above procedure 15 times for each network condition (capacity constraint and router queue size combination). 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 BDP router queue size] [GeForce, Luna game system].

A complete run of all iterations for both systems takes about 24 hours providing for performance that accounts for any time-of-day affects.

Table III provides a summary of the key experimental parameters.

TABLE III: Experimental parameters.

Game system	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

IV. ANALYSIS

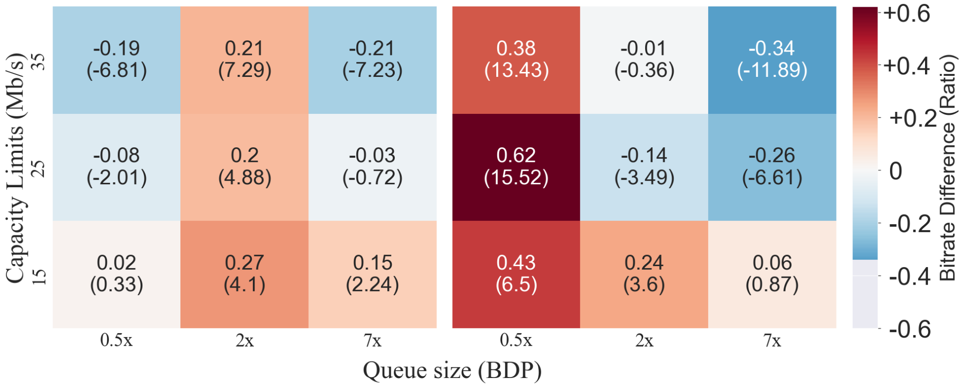
This section compares the different cloud-based game streaming systems with capacity and queue limits, considering: 1) game streaming bitrates and DASH video streaming bitrates (Section IV-A), and 2) bitrate fairness (Section IV-B).

A. *Bitrates*

We start analysis with a bitrate comparison (computed every 0.5 seconds) for both cloud-based game streaming systems 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 380s shows when the DASH flow stops.

Before the competing DASH flow arrives (i.e., up to time 190s), mostly the two 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. Overall, Luna exhibits lower bitrate variation, while GeForce tends to have higher bitrate variation.

When the DASH flows arrive (at 190s), the bitrates for both systems decrease, indicating they respond to the presence of



(a) Geforce (b) Luna

Fig. 3: Ratio of bitrate difference (i.e., difference ÷ capacity) for a game system competing with a DASH flow.

traffic competing for the available capacity and, similarly, the systems recover to their original bitrates after the DASH flows leave (after time 380s). When competing with DASH flows, the 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).

B. *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.

Figure 3 uses heatmaps to depict the results. There is one large box for each game system (GeForce on the left, 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, 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).

V. DISCUSSION

Previous work [1] examines cloud-based game streaming system response to competing bulk-downloads. The results show competing bulk-downloads impact game streams’ 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 (not shown due to space constraints). The competing DASH flows do not fare as well, typically getting less than half the capacity and frequent quality switches, although interrupts remain low.

In general, there are significant differences in congestion response across both 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 both 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.

VI. LIMITATIONS AND FUTURE WORK

Our focus is on the game player when 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 – may yield different results for both flows. 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 [3]. 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 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 other local- or metropolitan-area networks including 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 [15] or PIE [16]) might yield different results and could be considered for future study.

Our experiment setup allowed us to keep the same DASH configuration across runs, but commercial systems may behave differently, even while using DASH. Future work could involve using commercial DASH video such as Netflix or YouTube. Similarly, our current testing setup is limited to only two systems whereas there are other commercial systems in use that may behave differently. Future work could broaden testing to other platforms, such as Sony Playstation Now and Microsoft XCloud for comparative performance evaluation.

VII. CONCLUSIONS

Emerging cloud-based game streaming systems hold the promise of providing a convenient gaming experience for players as long as the network conditions can meet the challenge. In particular, streaming computer games need high definition frames sent at high 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 two commercial systems – 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 shows the two game systems have similar bitrates that operate near the capacity constraints, and even for constrained conditions (lower capacity, smaller queue size) none 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, and Luna takes 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 competing DASH flows mostly operate without interruptions in their playout, except for capacity constrained conditions, especially when competing with Luna.

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] X. Xu and M. Claypool, "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, 2022, p. 305–316.

[2] K. Spiteri, R. Sitaraman, and D. Sparacio, "From theory to practice: Improving bitrate adaptation in the dash reference player," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 15, no. 2s, jul 2019. [Online]. Available: https://doi.org/10.1145/3336497

[3] X. Xu and M. Claypool, "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, May 2021.

[4] M. Suznjevic, I. Slivar, and L. Skorin-Kapov, "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, 2016, pp. 1–6.

[5] M. Claypool, D. Finkel, A. Grant, and M. Solano, "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)*, 2012, pp. 1–6.

[6] M. Carrascosa and B. Bellalta, "Cloud-gaming: Analysis of Google Stadia Traffic," 2020.

[7] X. Xu and M. Claypool, "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, Jun. 2022.

[8] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *IEEE Communication Surveys Tutorial*, vol. 17, no. 1, pp. 469 — 492, Sep. 2014.

[9] M. Garcia, F. D. Simone, S. Tavakoli, N. Staelens, S. Egger, K. Brunnström, and A. Raake, "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, Sep. 2014.

[10] M. Garcia, D. Dytko, and A. Raake, "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, Sep. 2014.

[11] A. Bentaleb, B. Taani, A. Begen, C. Timmerer, and R. Zimmermann, "A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, 2019.

[12] D. Bhat, A. Rizk, and M. Zink, "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, Jun. 2017.

[13] J. Gettys and K. Nichols, "Bufferbloat: Dark Buffers in the Internet: Networks without Effective AQM May Again be Vulnerable to Congestion Collapse," *ACM Queue*, no. 11, pp. 40–54, Nov. 2011.

[14] Wikipedia contributors, "Tc (Linux)," 2020, [Accessed 27-Jan-2020]. [Online]. Available: https://en.wikipedia.org/wiki/Tc_(Linux)

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

[16] R. Pan, P. Natarajan, C. Piglione, M. S. Prabhu, , V. Subramanian, F. Baker, and B. VerSteeg, "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, Jul. 2013.