WPI-CS-TR-22-01 April 2022

Measurement of Network Congestion caused by Cloud-based Game Streaming

by

Xiaokun Xu and Mark Claypool

Measurement of Network Congestion caused by Cloud-based Game Streaming

Xiaokun Xu*
Mark Claypool*
Worcester Polytechnic Institute
Worcester, MA, USA

ABSTRACT

Cloud-based game streaming has emerged as a viable way to play games anywhere with a good network connection. While previous research has studied the network turbulence of game streaming traffic, there is as of yet no work exploring how cloud-base game streaming responds to rival connections on a congested network. This paper presents experiments measuring and comparing the network response for three popular commercial streaming services – Google Stadia, NVidia GeForce Now, and Amazon Luna – competing with TCP flows on a congested network. Analysis of the bitrates, loss and latency show that the three systems have marked different approaches to network congestion, but are mostly fair to competing TCP flows sharing a bottleneck link.

CCS CONCEPTS

• **Networks → Network measurement, performance analysis**.

KEYWORDS

games, streaming, TCP, network capacity, congestion control

ACM Reference Format:
Xiaokun Xu and Mark Claypool. 2022. Measurement of Network Congestion caused by Cloud-based Game Streaming. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 16 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The growth in cloud computing coupled with high-capacity networks has brought the opportunity for cloud-based game systems that stream game content as video down to the player. Many companies have capitalized on this opportunity and introduced game-streaming services such as Sony PlayStation Now, Microsoft xCloud, Google Stadia, NVidia GeForce Now, and Amazon Luna. The cloud-based game streaming market is growing rapidly with a value of $865.8 million USD in 2021 and is expected to expand at a compound annual growth rate of 48.2% from 2021 to 2027 [11].

Cloud-based game streaming differs from traditional computer games in that game clients do not run full versions of the game engine. Instead, only the cloud-based server handles the game engine logic -– applying physics to game objects, resolving collisions, processing AI, etc. -– and renders the game frames, streaming the game as video to the game client. This allows the game client to be relatively lightweight, mostly just playing the streamed game frames and sending player input to the server. However, the significant disadvantages of cloud-based game streaming are the increased traffic required for the game frame streaming and the added round-trip latency for all player actions. In particular, the bitrate requirements for frequent, high-quality video frames can cause congestion, decreasing player quality of experience and impacting co-located network traffic.

Prior work has shown that cloud-based game streaming requires a high bandwidth network and is sensitive to network latency [1, 6, 13]. While studies have analyzed network traffic for specific cloud systems like Google Stadia, NVidia GeForce Now and Sony PSNow [8, 10, 19], lacking are comparative aspects across systems, especially how cloud-based game streams respond to congestion. 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.

This work presents an analysis of the network congestion caused by three commercial cloud-based game streaming systems – Google Stadia, NVidia GeForce Now and Amazon Luna – providing a direct comparison of their bitrate use over time and impact on network congestion when competing for scarce bitrates with TCP flows. To do so, we designed a novel methodology that runs the game systems via a script, with the games played automatically to ensure similar player actions across runs. The network testbed and experiments are done over the Internet, but are designed to be as comparable across runs as possible by interlacing runs of each game system serially to minimize temporal differences, and by doing 15 runs for each test condition to provide for a large sample.

The results show the three game systems use similar bitrates when faced with capacity constraints, with Google Stadia having the most bitrate variance. When competing for network capacity with bulk-download TCP flows, Google Stadia and Amazon Luna share the capacity fairly equally, but NVidia GeForce Now defers to the competing flow more than necessary. The degree to which fairness is achieved depends upon the network bottleneck capacity and the bottleneck queue size, with more packet loss and more unfairness caused by more congested conditions – i.e., tighter capacity limits and smaller queue sizes have more unfairness.

*Both authors contributed equally to this research.

Conference'17, July 2017, Washington, DC, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

The rest of this paper is organized as follows: Section 2 provides related work on the network aspects of cloud-based game streaming; Section 3 describes our methodology, including testbed setup and experiment design and parameters; Section 4 analyzes the experimental results; Section 5 mentions limitations and future work, and Section 3.1 summarizes our conclusions.

2 RELATED WORK

There are studies analyzing the network performance of early commercial cloud-based game systems, such as OnLive [18] and Gaikai [17]. Manzano et al. [13] 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. [6] 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. [15] measure network traffic for NVidia GeForce Now and find GeForce requires bitrates significantly higher than earlier systems (about 25 Mb/s today compared to 6 Mb/s previously). Xu et al. [19] 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).

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 network flows, nor do they compare systems to each other.

The closest work to our own that we are aware of is from Marc et al. [4] that limits link capacities for Google Stadia during gameplay, finding Stadia adjusts the resolution and/or frame rate in response to a bitrate reduction. However, the experiments conducted do not necessarily represent responses TCP flows competing for the network, nor are other, non-Stadia systems considered and compared as does our work.

3 METHODOLOGY

To capture the network congestion caused by cloud-based games, we selected 3 popular commercial cloud-based game streaming platforms and a game common to all (Section 3.1), setup a measurement testbed that allowed for controlling congestion conditions (Section 3.3), gathered network traces (Section 3.4), and analyzed the data (Section 4).

3.1 Platform and Game Selection

We selected 3 cloud-based game streaming platforms – Google Stadia, NVidia GeForce Now, and Amazon Luna – based on the their current and likely future popularity for game players. While Luna and GeForce offer native applications for client-side player, all three support play via the Google Chrome browser which we used for a fair comparison across systems.

For game selection, as for the platform, we sought a game that could be played on each 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.

Since gameplay visuals (i.e., what the player sees and is streamed to the client) depends upon the player's actions, we developed innovative scripts to play the game automatically, thus providing identical, repeatable gameplay conditions across runs and across platforms. Our scripts automatically open the game (with input appropriate for each platform), select the right level, and then play the game automatically as if run by a human player.

3.2 Network Conditions

Our goal is to assess the response to congestion for the cloud-based game streaming systems considering congestion arising from both network capacity limits and competing traffic. The network capacity limits allow comparison of system response to self-induced congestion arising from various "last-mile" network conditions provided, say, by an Internet Service Provider (ISP). Competing traffic allows comparison of system congestion response 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) 30% below the maximum capacity required.

The dynamics of many congestion control algorithms (e.g., TCP) is influenced by the size of the queue at the bottleneck router [7]. A general rule of thumb is that the bottleneck buffer queue size should be a multiple of the bottleneck capacity and the round-trip delay, otherwise known as the bandwidth-delay product (BDP). Guidelines suggest a "good" queue size is the $(BDP)/sqrt(n)$, 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" [9]. We consider queue sizes that are: 1) shallow, at about one-half the BDP, 2) typical, at about 2x the BDP, and 3) bloated, at about 8x the BDP.

3.3 Measurement Testbed

Figure 1 depicts the general setup for our measurement testbed. Our testbed automatically plays the game 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 or Amazon Luna). For experiments with competing traffic, the bottleneck link (from the router down to the clients) is shared by an iperf client that does bulk-downloads from an iperf server. The game client has 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 Gb/s Ethernet NIC. The PC has an LED monitor with 1920x1080 pixels running at 60 Hz. The iperf client and server are both Alienware PCs with an 8-core Intel i7-4790K CPU @ 4 GHz with 16 GB RAM running Ubuntu 20.04 LTS, Linux kernel version 5.4, and connect with Gb/s Ethernet NICs.

The Game client and iperf client PCs connect via a 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

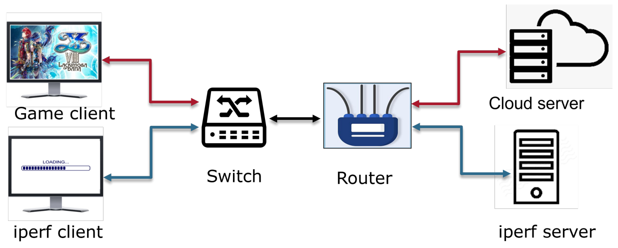


Figure 1: Measurement testbed.

20.04 LTS, Linux kernel version 5.4, using `tc` [16] and `netem` to constrain the network capacity. The router also runs `Wireshark` to gather network traces.

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 downstream bitrates from the campus network, through the router and to the client PC of 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.

3.4 Experiments

Our pilot studies determined 3 minutes of gameplay provided for the full range of observed network behaviors. Thus, for all analysis, the Wireshark traces were trimmed to 3 minutes of core gameplay (i.e., loading, menus, etc. are *not* included — just gameplay).

The average bitrates for the 3 platforms and our selected game on an unconstrained network were around 25 Mb/s. Thus, for our experiments, we tested 3 network conditions: a "good" connection with a capacity limit of 35 Mb/s, a "normal" connection with a capacity limit of 25 Mb/s, and a "bad" connection with a capacity limit of 15 Mb/s.

Additional scripts automatically launch a `ping` command to the game server, and connect to the router to set the queue size and bottleneck capacity limit, as appropriate, and launch Wireshark. Again, we trim all data analysis to only the core gameplay.

For each round, the fully-automated experiment procedure is:

(1) Start the game in the browser and wait for the game load.
(2) Connect to the router to set the bitrate and queue size, and start Wireshark.
(3) Initiate a `ping` to the game server.
(4) Run the script on the game client that plays the game for 3 minutes.
(5) For appropriate runs, after 3 minutes, start iperf on the iperf client,
(6) Continue the script which plays the game for 3 minutes.
(7) Close the game, all data collection tools, and reset the router to the unconstrained conditions.
(8) Repeat the above procedure for each of the three systems (Stadia, GeForce and Luna).

We repeat the above 15 times for each network condition and each platform. Since Internet conditions from the campus network to the game servers can change over time, for each setting, note that we stripe across game service to keep system comparisons as temporally close as possible. All this is done by the scripts automatically, without manual intervention for consistency.

A complete run of all systems and all iterations takes about 24 hours providing for performance that accounts for any time-of-day affects. Data was gathered for a weekday in December 2021.

Table 1 provides the full list of experimental parameters.

Table 1: Experimental parameters.

Game	Ys VIII: Lacrimosa of Dana
Capacity limit	15, 25, 35 Mb/s, or no restriction
Queue size	0.5x, 2x, 8x BDP
Competing connection	none and iperf TCP flow
Trace length	3 minutes
Iterations	15 runs per game system, per condition

4 ANALYSIS

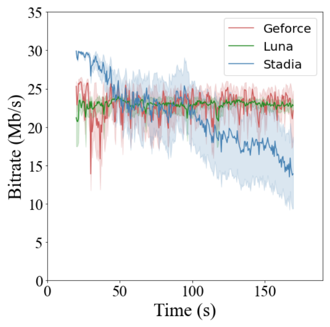
This section compares the different cloud-based game streaming systems with capacity limits and competing TCP flows considering: 1) bitrates (Section 4.1) and 2) delay (round-trip time) and packet loss (Section 4.2). This section then summarizes bitrate fairness across systems and network conditions when competing with a TCP flow for capacity (Section 4.3).

4.1 Bitrates

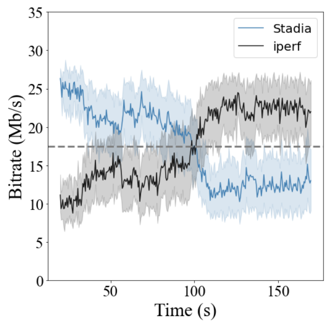
We start analysis with a bitrate comparison (computed every 0.5 seconds) of each gaming system when they are not competing with another TCP flow (i.e., no competing iperf flow) for our three different capacity constraints – 35 Mb/s, 25 Mb/s and 15 Mb/s – each with a typical 2x BDP queue. Figure 2 depicts the results, with the left side (sub-figures a, e and i) showing bitrates without iperf. For each graph, the x-axis is gameplay time, in seconds, and the vertical axis is the measured bitrate, in Mb/s. The mean for each system is showed with a colored line (Stadia - blue, GeForce - red, Luna - green) with the shading depicting 95% confidence intervals for that point across the 15 runs. Note, the graphs show the three systems at once for comparison, but they were run independently, not simultaneously.

From the graphs, when there is a surfeit of capacity (35 Mb/s) the systems use a similar bitrate. However, Stadia has considerably more bitrate variance and an overall downward trend over the 3 minute run. For constrained capacities (25 Mb/s and 15 Mb/s), Luna and GeForce settle near the capacity limit, but Stadia stays considerably under this limit and still has the downward trend. Periodic increases in Stadia bitrates (e.g., 75s, 125s and 175s in Figure 2e) suggest it is probing for available bandwidth, before returning at the previous bitrate.

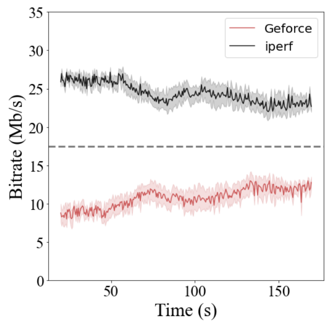
Figure 3 depicts another representation of this same bitrate data, here shown with a boxplot. The x-axis has 3 sets of a capacities (15, 25 and 35 Mb/s) and the y-axis the bitrate. Each box depicts quartiles and median for the distribution using the same colors as before (blue for Stadia, orange/red for GeForce, and green for Luna). The whiskers span from the minimum to the maximum and the white circles show the average values. From the graph, the large



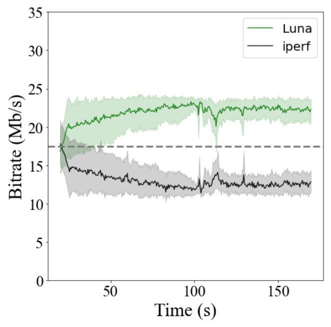
(a) B35 without iperf



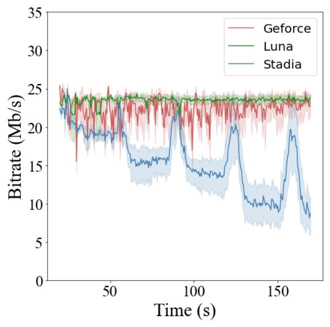
(b) B35 Stadia and iperf



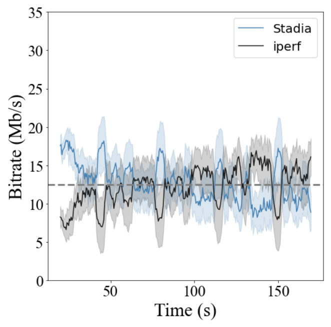
(c) B35 Geforce and iperf



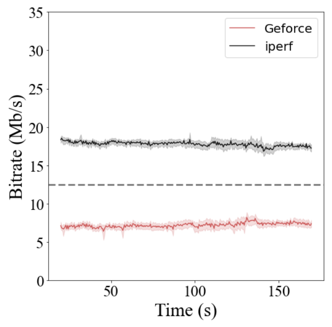
(d) B35 Luna and iperf



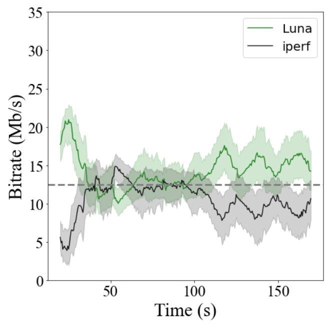
(e) B25 without iperf



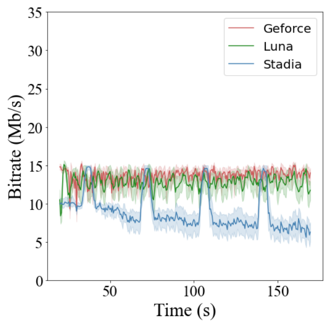
(f) B25 Stadia and iperf



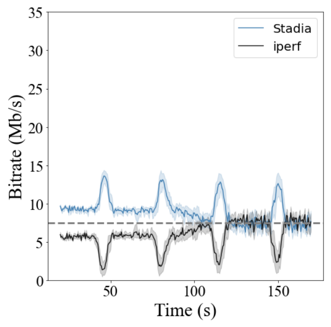
(g) B25 Geforce and iperf



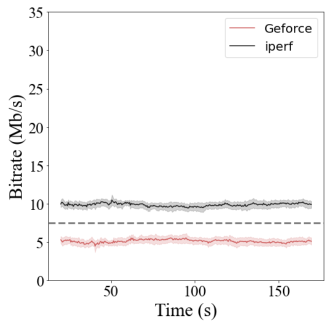
(h) B25 Luna and iperf



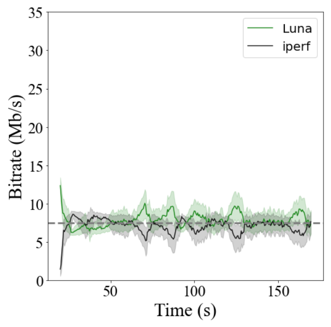
(i) B15 without iperf



(j) B15 Stadia and iperf



(k) B15 Geforce and iperf



(l) B15 Luna and iperf

Figure 2: Bitrate versus time with different capacity limits.

bitrate variation for Stadia is apparent from the vertical height of its boxes, as is the generally lower bitrates for Stadia for the 15 and 25 Mb/s capacity limits.

We next analyze bitrates when competing with an iperf flow. Using TCP, iperf expands to consume the available bandwidth, but responds to packet losses as indicators of congestion. Given that there are two flows for these experiments (one game-system flow and one iperf flow), a fair share of the bottleneck bandwidth would be one-half the capacity limit. The graphs in the 2nd, 3rd and 4th columns of Figure 2 depict bitrates for the simultaneous flows for each system in a column (Stadia, GeForce and Luna) and each capacity (35 Mb/s, 25 Mb/s and 15 Mb/s) as a row. The lines again show average bitrate with 95% confidence intervals. The horizontal dashed line depicts the fair share at 1/2 the capacity limit.

From the graphs, the 9 conditions depicted all look quite different. GeForce stands out visually as deferring to the iperf flow, with the iperf average being consistently above the fair share line and GeForce being below. In contrast, Stadia and Luna tend to share the capacity evenly with iperf, with averages for both flows near the dashed line. The exception is Luna at 35 Mb/s where the Luna flow is almost entirely above the fair share and iperf is below. As for the earlier graphs, Stadia has considerably more variation in bitrate than the other systems, and the periodic bitrate "probing" is quite visible in the 15 Mb/s condition (Figure 2j).

Figure 4 shows a boxplot of the bitrate differences between each system and the competing iperf flow. The axes and boxes are as for Figure 3, but here the values are the game system flows minus the iperf flows computed each 0.5 second. Values at 0 (the horizontal line) are completely fair in that both flows have equal bitrates. Values above 0 mean the streaming system has more of the capacity than the iperf flow and values below 0 mean the iperf flow has more of the capacity than the streaming system flow. From this graph, it

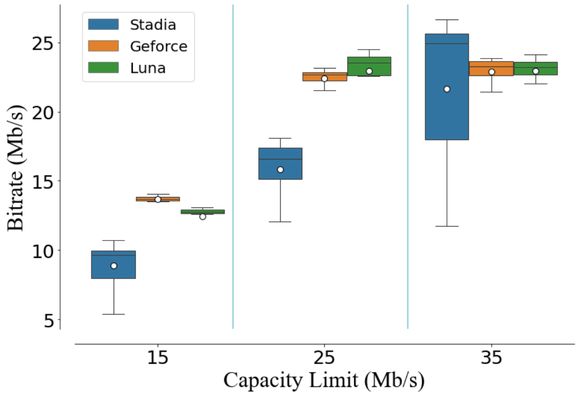


Figure 3: Bitrate distributions.

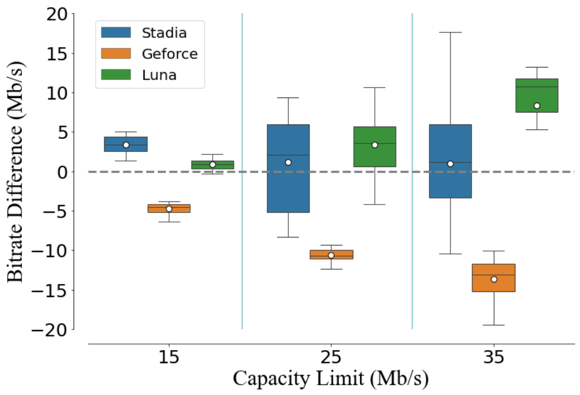


Figure 4: Bitrate difference (system - iperf) distributions.

is apparent that GeForce defers to the iperf flow nearly all the time, while Luna rarely does and has significantly more of the capacity for the higher 25 and 35 Mb/s limits. Stadia generally has slightly more of the capacity than the iperf flow (and nearly always does at 15 Mb/s) but there are many cases where iperf has a higher bitrate.

4.2 Delay and Packet loss

Indicators of congestion are delay (which manifests when the router queue becomes filled with excess traffic) and packet loss (which happens when a full router queue drops incoming packets).

Table 2 (without iperf) and Table 3 (with iperf) have the round trip times for the 3 systems under each network condition (capacity and queue size). Each value is the mean for the 3 minutes of gameplay with standard deviations shown in parentheses. In the 35 Mb/s condition without iperf, Geforce has lowest round-trip time from our campus network (about 4 ms), followed by Stadia (about 12 ms) and Luna (about 16 ms). These values increase with a decrease in capacity, but do not reach even 0.5x BDP suggesting the systems themselves to not look to saturate available capacity until there is loss. With iperf, the round-trip times are consistently at the limit dictated by the queue size, which makes sense given that the iperf flow generally increases sending rates until there is packet loss.

Table 2: Round-trip time (ms) without iperf.

	BDP 0.5x			BDP 2x			BDP 8x		
Capacity	Stadia	GeForce	Luna	Stadia	GeForce	Luna	Stadia	GeForce	Luna
15 Mb/s	11.9 (1.1)	4.1 (0.9)	16.7 (2.1)	13.9 (6.5)	7.6 (8.4)	20.2 (9.9)	16.4 (12.6)	7.8 (8.6)	18.3 (14.2)
25 Mb/s	11.9 (1.5)	4.1 (0.9)	15.9 (0.8)	13.7 (6.4)	5.8 (5.1)	15.9 (0.8)	17.7 (14.4)	6.4 (7.6)	15.9 (10.9)
35 Mb/s	11.6 (0.7)	4.1 (0.8)	16.1 (0.8)	11.6 (0.8)	3.9 (0.7)	16.0 (0.9)	11.6 (0.8)	6.9 (10.9)	16.1 (2.8)

Table 3: Round-trip time (ms) with iperf.

	BDP 0.5x			BDP 2x			BDP 8x		
Capacity	Stadia	GeForce	Luna	Stadia	GeForce	Luna	Stadia	GeForce	Luna
15 Mb/s	13.3 (1.9)	5.5 (1.7)	17.1 (2.1)	28.5 (4.7)	25.1 (4.2)	37.6 (5.9)	84.3 (15.9)	84.0 (10.2)	120.5 (15.8)
25 Mb/s	13.3 (1.9)	6.1 (1.8)	17.1 (2.2)	27.7 (5.3)	21.2 (3.5)	37.9 (6.3)	89.3 (10.8)	86.3 (9.2)	121.4 (17.9)
35 Mb/s	13.4 (2.1)	6.2 (1.8)	17.0 (2.1)	27.3 (5.2)	20.9 (3.6)	37.9 (5.5)	82.9 (15.7)	87.4 (13.9)	122.9 (13.0)

Table 4: Packet loss (percent) without iperf.

	BDP 0.5x			BDP 2x			BDP 8x		
Capacity	Stadia	GeForce	Luna	Stadia	GeForce	Luna	Stadia	GeForce	Luna
15 Mb/s	0.04 (0.07)	0.04 (0.09)	0.06 (0.06)	0.04 (0.05)	0.04 (0.05)	0.04 (0.05)	0.04 (0.05)	0.02 (0.04)	0.04 0.05)
25 Mb/s	0.09 (0.07)	0.03 (0.05)	0 (0)	0.05 (0.05)	0.03 (0.03)	0 (0)	0.04 (0.03)	0.003 (0.01)	0 (0)
35 Mb/s	0 (0)	0.03 (0.06)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)

Table 5: Packet loss (percent) with iperf.

	BDP 0.5x			BDP 2x			BDP 8x		
Capacity	Stadia	Geforce	Luna	Stadia	Geforce	Luna	Staida	Geforce	Luna
15 Mb/s	0.24 (0.01)	0.05 (0.09)	0.13 (0.11)	0.17 (0.09)	0.03 (0.03)	0.04 (0.08)	0.06 (0.09)	0.02 (0.03)	0.02 (0.04)
25 Mb/s	0.18 (0.10)	0.05 (0.13)	0.11 (0.09)	0.11 (0.09)	0.02 (0.03)	0.04 (0.03)	0.04 (0.05)	0.01 (0.03)	0.01 (0.03)
35 Mb/s	0.14 (0.07)	0.01 (0.04)	0.06 (0.06)	0.04 (0.05)	0 (0)	0.01 (0.03)	0.03 (0.05)	0 (0)	0.01 (0.02)

This illustrates how large router queues in the presence of competing flows during congestion mean added delay for game-streaming system. This delay has been shown to degrade player performance and quality of experience [5].

Table 4 (without iperf) and Table 5 (with iperf) show the corresponding loss rates (percent) for each condition. When there are no competing flows (i.e., without iperf), the systems themselves induce almost no packet loss, even for constrained conditions. However, with iperf, packet loss rates are noticeably higher, but still under a 0.5% in all cases, even the most bitrate constrained.

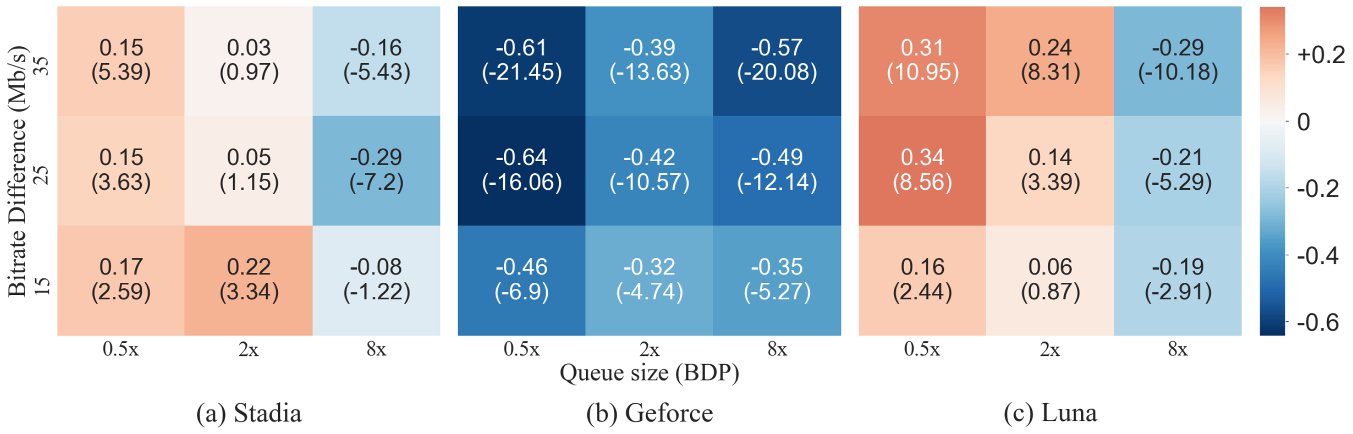


Figure 5: Bitrate difference (system-iperf) heatmap.

4.3 Summary

We summarize the bitrate difference between each game system and iperf for each network condition via a heatmap in Figure 5. For capacity limit x_b, cloud streaming bitrate x_1 and iperf bitrate x_2, the bitrate difference is:

$$f(x_1, x_2) = \frac{x_1 - x_2}{x_b}$$

There is one large box for each 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 as rows and 0.5x, 2x and 8x BDP queues as columns. The numbers in the boxes are average difference in throughput for the game system minus the competing iperf flow, shown normalized by the capacities (ranging from -1 to 1) with absolute differences in parentheses. The warm, red tones show where the game system has a higher bitrate and the cool, blue tones where the game system has a lower bitrate. Visually, GeForce is entirely "cool" and always gets less than it's fair share of the capacity when competing with iperf. In contrast, Stadia and Luna have some "hot" areas where they get more than their fair share of the capacity, with Luna having the "hottest" areas for small queues (0.5x) BDP and high capacities (25 and 35 Mb/s). However, both Stadia and Luna are "cool" with less than their fair share of the bandwidth for large queue sizes (8x BDP).

We also compute Jain's fairness index [2], one of the most widely used metrics for measuring the fairness of system resources allocation. For two flows with throughputs x_1 and x_2, Jain's fairness is computed as:

$$f(x_1, x_2) = \frac{(x_1 + x_2)^2}{2(x_1^2 + x_2^2)} \tag{1}$$

and ranges from $\frac{1}{2}$ (most unfair) to 1 (most fair). Figure 6 shows the fairness results as a heatmap corresponding to the same conditions as for Figure 5. The green shades indicate better fairness, whereas red shades indicates worse fairness. Generally, Stadia and Luna are fair (mostly green) whereas GeForce has more red (iperf getting more of the capacity). The lowest capacity (15 Mb/s) tends to have the most fairness, with slightly better fairness for higher queue sizes (2x and 8x).

5 LIMITATIONS AND FUTURE WORK

There are several limitations to our work, that also suggest future work.

Our experiments are for one game only and prior work has shown that the bitrates for different games on the same system can vary considerably [19]. Future work should see if the comparative differences illustrated here hold for other games, as well.

All the experiments with iperf use the Cubic [12] congestion control algorithm for TCP. While Cubic is the default for Linux, emerging congestion control protocols such as the Bottleneck Bandwidth Round-trip time (BBR) [3] have different responses to congestion and queue sizes [7]. Future work should examine BBR and perhaps other congestion control protocols.

Our router uses only a drop-tail queue, whereas prior research has shown considerably different behavior for Active Queue Management approaches (AQM) that signal congestion earlier in an effort to avoid full queues. Future work might consider experiments with the Controlled Delay (CoDel) AQM approach [14].

6 CONCLUSIONS

Emerging cloud-based game streaming systems promise to provide for a convenient gaming experience for players, provided that network conditions are sufficient. In particular, streaming computer games need high definition frames sent at high frame rates (at least 60 f/s). This, in turn, requires high bitrates that have the potential to congestion last mile networks, particularly when competing for capacity with other network flows. This paper compares three commercial systems – Google Stadia, NVidia GeForce and Amazon Luna – with repeated runs of the same game across capacity constraints and bottleneck queue sizes, while competing for capacity with a TCP flow.

Analysis of the results show the three systems have typical bitrates when there is more than the maximum bitrate they require (e.g., 35 Mb/s), but when bitrates are constrained, Stadia is the

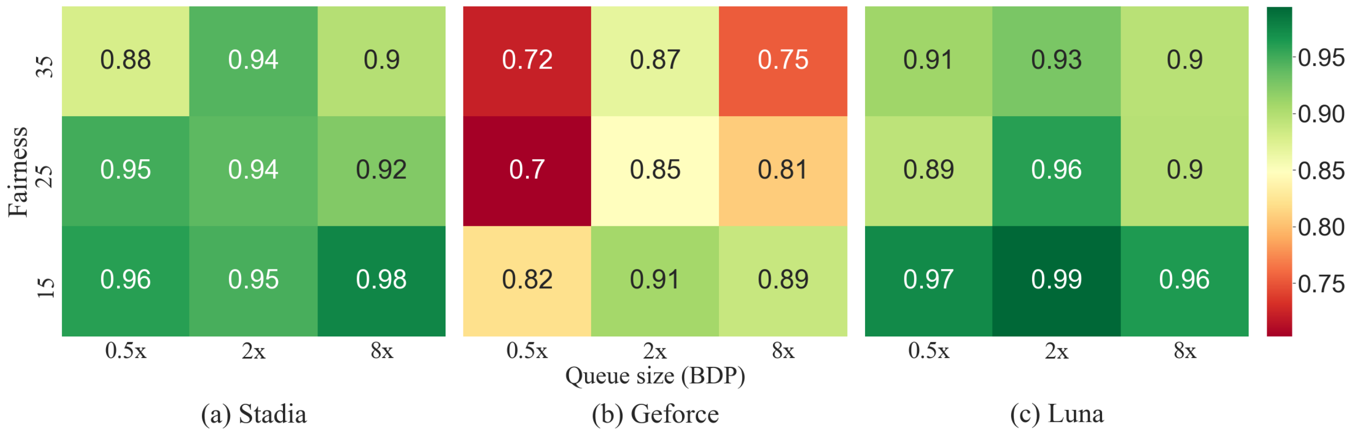


Figure 6: Fairness heatmap.

most conservative and decreases bitrates the most, well below the capacity limit. When competing with a bulk-download TCP flow, Stadia and Luna generally share the available capacity fairly, but GeForce defers and lets the TCP flow have more than half of the capacity. Large bottleneck queues (buffer bloat) tend to favor the competing TCP flow and result in larger delays for the game systems (which is bad for user quality of experience). More constrained network conditions are generally more fair, although they also have the highest loss rates, albeit all are under 0.5%. These results provide a better understanding of game systems interaction with constrained and competitive network links, and should be useful to better plan for, and hopefully deter, resulting network congestion.

Future work can include gathering framerate data and analyzing how framerate changes in different scenarios. Or, select more games and cloud gaming systems to do the experiment which make the results more generalized. Another interesting and meaningful idea is doing the same experiment with iperf with BBR congestion control algorithm to see how cloud gaming system interact with BBR, and compare the results with the experiments we have done.

REFERENCES

[1] 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. https://doi.org/10.1109/HAVE.2013.6679619

[2] Per Nikolaj D. Bukh. 1992. *Interfaces* 22, 4 (1992), 113–115. http://www.jstor.org/stable/25061650

[3] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and Van Jacobson. 2017. BBR: Congestion-based Congestion Control. *Commun. ACM* 60, 2 (Jan. 2017), 58–66.

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

[5] 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), 480–495. https://doi.org/10.1109/TMM.2013.2291532

[6] 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. https://doi.org/10.1109/NetGames.2012.6404013

[7] Saahil Claypool, Mark Claypool, Jae Chung, and Feng Li. 2019. Sharing but not Caring - Performance of TCP BBR and TCP CUBIC at the Network Bottleneck. In *Proceedings of the 15th IARIA Advanced International Conference on Telecommunications (AICT)*. Nice, France.

[8] 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), 247–260. https://doi.org/10.3390/network1030015

[9] 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.

[10] 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. https://doi.org/10.23919/CNSM52442.2021.9615562

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

[12] Sangtae Ha, Injong Rhee, and Lisong Xu. 2008. CUBIC: A New TCP-Friendly High-Speed TCP Variant. *ACM SIGOPS Operating Systems Review* 42, 5 (2008).

[13] 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. https://doi.org/10.1109/NetGames.2012.6404021

[14] K. Nichols, V. Jacobson, A. McGregor, and J. Iyengar. 2018. Controlled Delay Active Queue Management. *IETF. Request for Comments (RFC) 8289* (Jan. 2018).

[15] 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 *In Proceedings of the 8th International Conference on Quality of Multimedia Experience (QoMEX)*. Lisbon, Portugal, 1–6. https://doi.org/10.1109/QoMEX.2016.7498968

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

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

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

[19] 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.

APPENDICES

Sections are:

- A Frame rates for game systems measured with and without competing iperf flows.
- B Bitrates versus time with capacity limits and with competing TCP flow (CUBIC)
- C Bitrates versus time with capacity limits and with competing TCP flow (BBR)
- D Bitrates with 4 different games (Stadia and GeForce only)

Appendix A FRAME RATES

Frame rates captured on the client via presentmon.[1]

Table 6: Frame rate (f/s) without iperf.

	BDP 0.5x			BDP 2x			BDP 8x		
Capacity	Stadia	GeForce	Luna	Stadia	GeForce	Luna	Stadia	GeForce	Luna
15 Mb/s	59.9 (0.1)	59.5 (0.1)	59.9 (0.1)	59.89 (0.2	59.7 (0.2)	59.7 (0.2)	59.8 (0.2)	59.8 (0.2)	59.8 (0.3)
25 Mb/s	58.3 (0.9)	59.1 (0.2)	59.7 (0.4)	59.4 (0.3)	58.7 (0.4)	59.3 (0.1)	59.8 (0.2)	59.8 (0.2)	59.8 (0.2)
35 Mb/s	59.1 (0.5)	59.1 (0.2)	59.5 (0.8)	59.4 (0.4)	58.8 (0.2)	59.1 (0.2)	59.6 (0.2)	59.3 (0.2)	59.6 (0.3)

Table 7: Frame rate (f/s) with iperf.

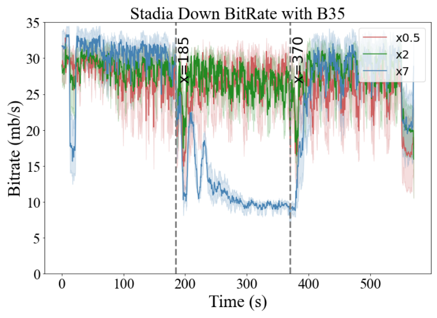
	BDP 0.5x			BDP 2x			BDP 8x		
Capacity	Stadia	GeForce	Luna	Stadia	GeForce	Luna	Stadia	GeForce	Luna
15 Mb/s	54.6 (2.1)	55.0 (1.8)	57.7 (0.7)	59.6 (0.5)	58.8 (0.3)	59.3 (0.1)	57.4 (0.7)	58.3 (0.3)	58.9 (0.5)
25 Mb/s	52.5 (0.9)	56.1 (1.1)	57.3 (1.4)	59.6 (0.3)	59.2 (0.3)	59.6 (0.3)	56.2 (0.4)	57.0 (0.7)	58.1 (0.5)
35 Mb/s	52.2 (1.3)	54.6 (0.6)	55.4 (1.9)	59.1 (0.3)	58.9 (0.4)	59.6 (0.3)	55.4 (0.6)	56.7 (0.8)	58.1 (0.5)

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

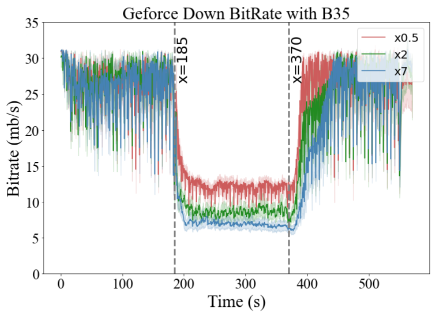
Appendix B BITRATES WITH CAPACITY LIMITS FOR CUBIC

Cloud systems: Stadia, GeForce, Luna
TCP CCA: Cubic
Bitrate constraints 35 Mb/s, 25 Mb/s, 15 Mb/s
Queue size: 0.5x, 2x, 7x BDP
Game: *Ys VIII: Lacrimosa of Dana* (Nihon Falcom, 2016).

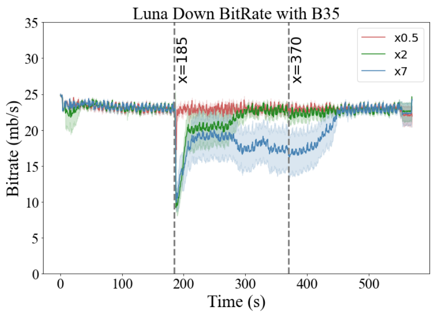
For each condition, the game is run without by itself for the first 3 minutes, then a competing iperf TCP flow is started and run for the next 3 minutes, then the iperf flow is stopped and the game run for the final 3 minutes. This process is repeated 15 times for each condition. Means are computed with 95% confidence intervals.



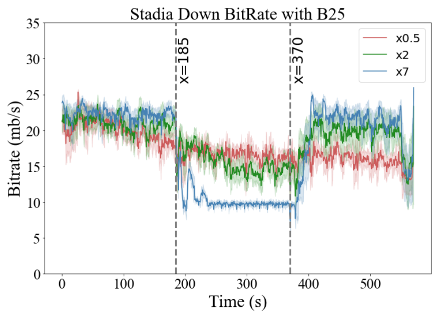
(a) B35 Stadia with iperf



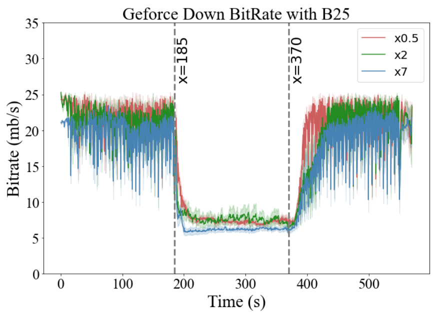
(b) B35 Geforce with iperf



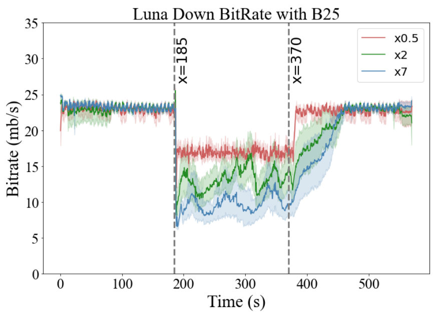
(c) B35 Luna with iperf



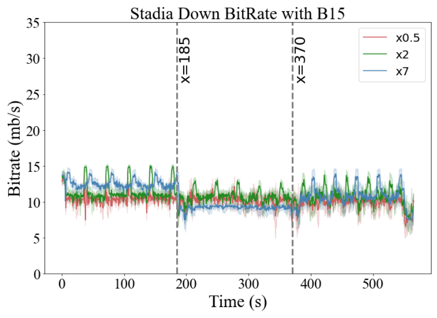
(d) B25 Stadia with iperf



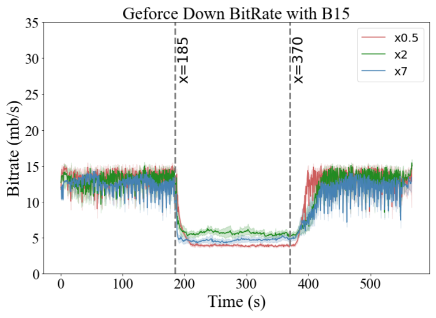
(e) B25 Geforce with iperf



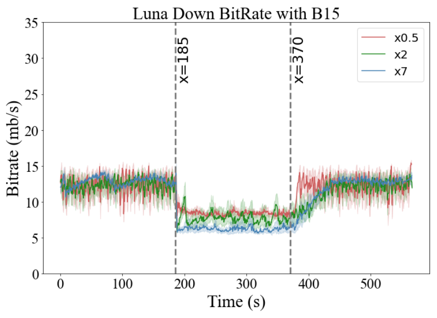
(f) B25 Luna and iperf



(g) B15 Stadia with iperf

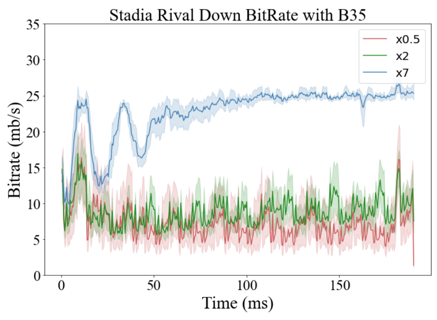


(h) B15 Geforce with iperf

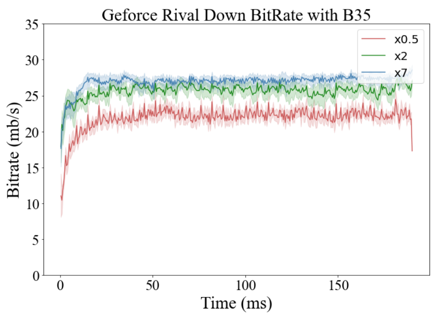


(i) B15 Luna with iperf

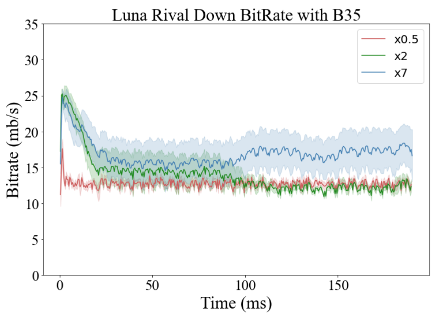
Figure 7: Bitrate versus time with different capacity limits.



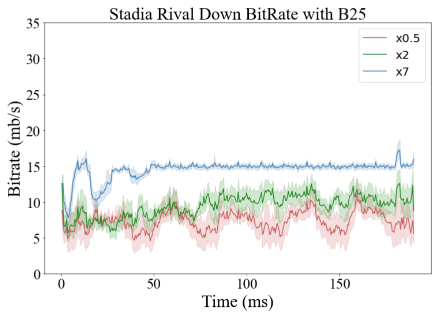
(a) B35 iperf with Stadia



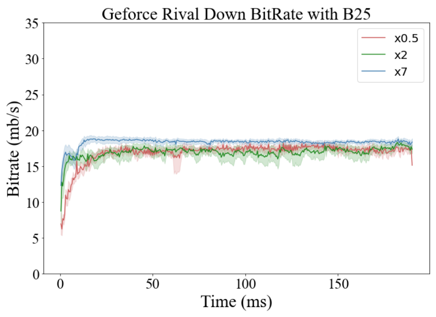
(b) B35 iperf with Geforce



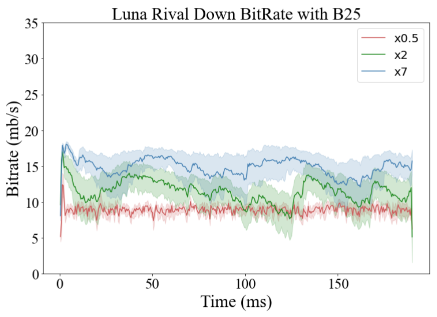
(c) B35 iperf with Luna



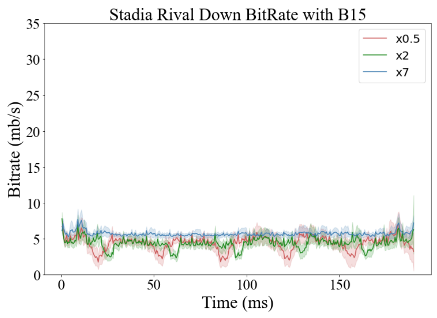
(d) B25 iperf with Stadia



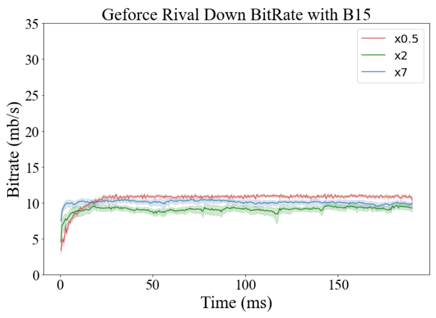
(e) B25 iperf with Geforce



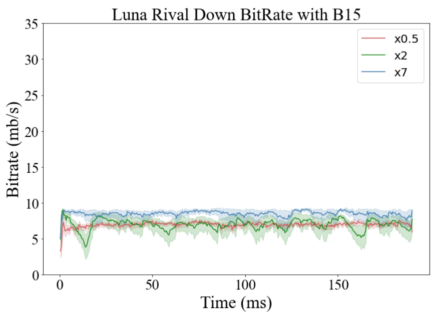
(f) B25 iperf with Luna



(g) B15 iperf with Stadia



(h) B15 iperf with Geforce



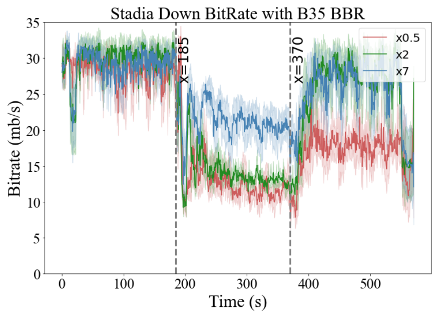
(i) B15 iperf with Luna

Figure 8: Bitrate versus time with different capacity limits.

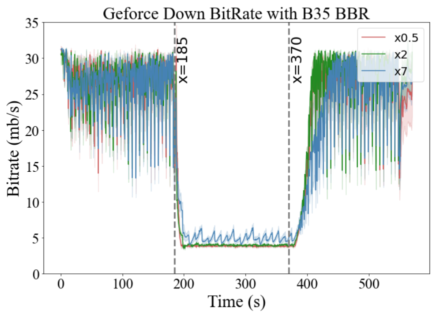
Appendix C BITRATES WITH CAPACITY LIMITS FOR BBR

Cloud systems: Stadia, GeForce, Luna
TCP CCA: BBR
Bitrate constraint: 35 Mb/s, 25 Mb/s, 15 Mb/s
QUeue size: 0.5x, 2x, 7x BDP
Game: *Ys VIII: Lacrimosa of Dana* (Nihon Falcom, 2016)

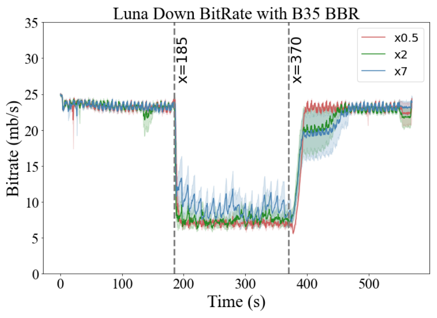
For each condition, the game is run without by itself for the first 3 minutes, then a competing iperf TCP flow is started and run for the next 3 minutes, then the iperf flow is stopped and the game run for the final 3 minutes. This process is repeated 15 times for each condition. Means are computed with 95% confidence intervals.



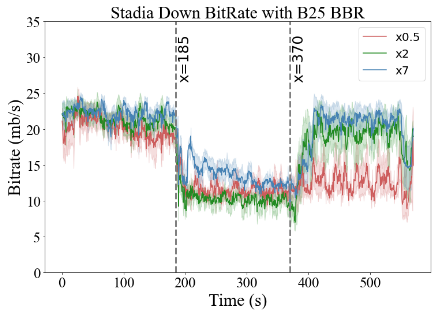
(a) B35 Stadia with iperf



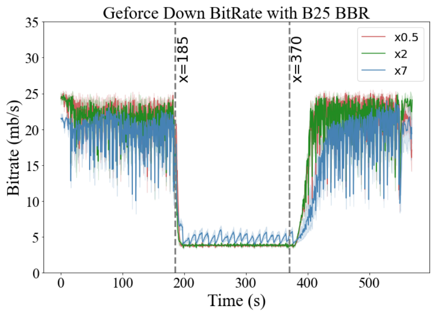
(b) B35 Geforce with iperf



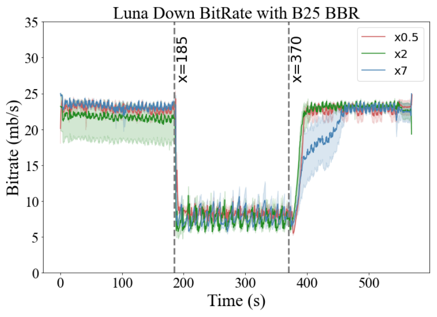
(c) B35 Luna with iperf



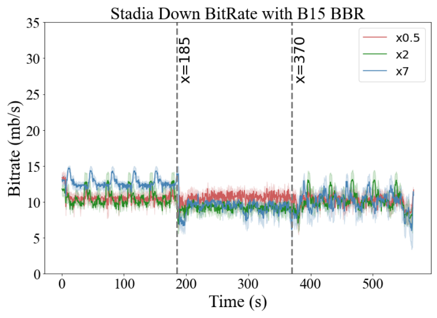
(d) B25 Stadia with iperf



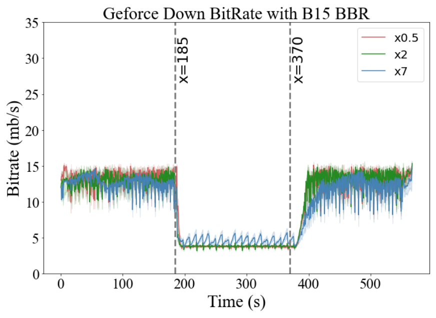
(e) B25 Geforce with iperf



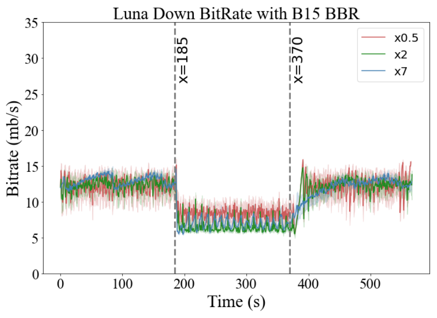
(f) B25 Luna and iperf



(g) B15 Stadia with iperf

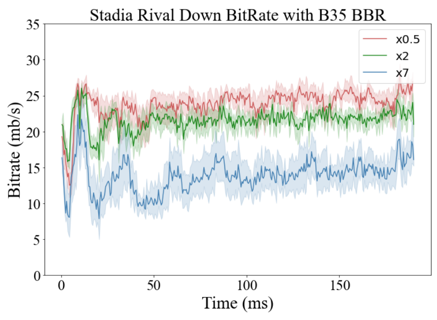


(h) B15 Geforce with iperf

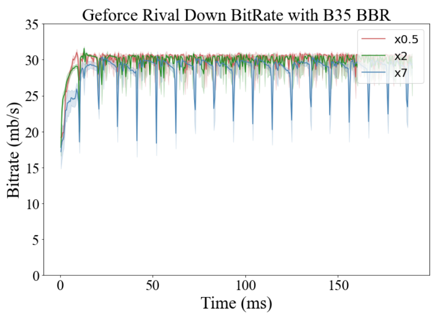


(i) B15 Luna with iperf

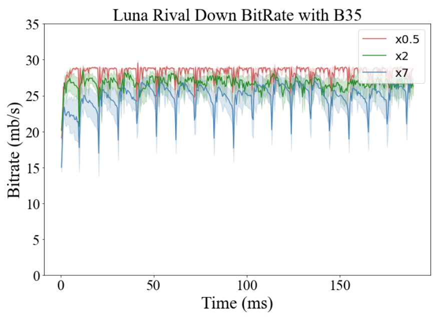
Figure 9: Bitrate versus time with different capacity limits (BBR).



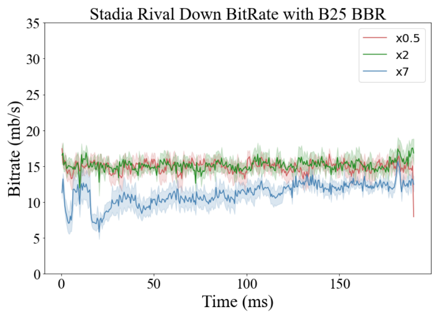
(a) B35 iperf with Stadia



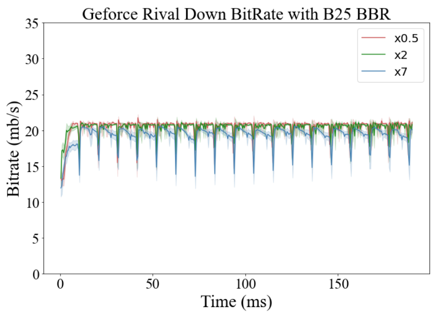
(b) B35 iperf with Geforce



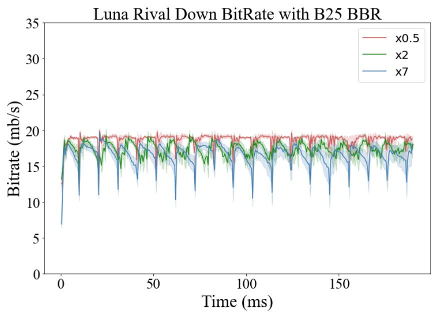
(c) B35 iperf with Luna



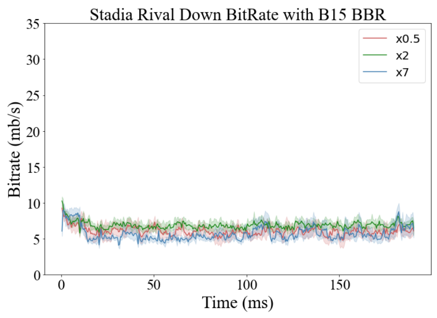
(d) B25 iperf with Stadia



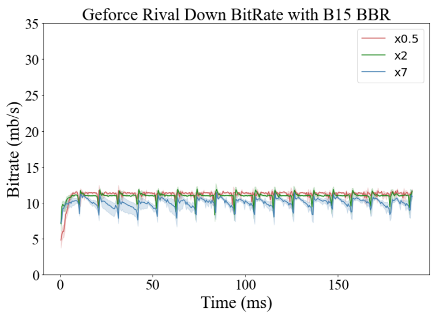
(e) B25 iperf with Geforce



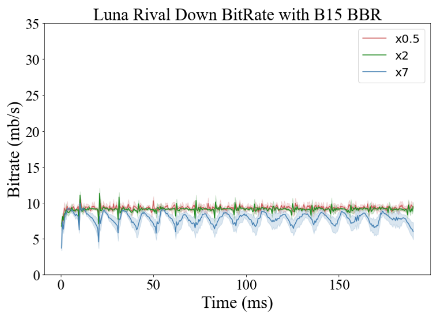
(f) B25 iperf with Luna



(g) B15 iperf with Stadia



(h) B15 iperf with Geforce



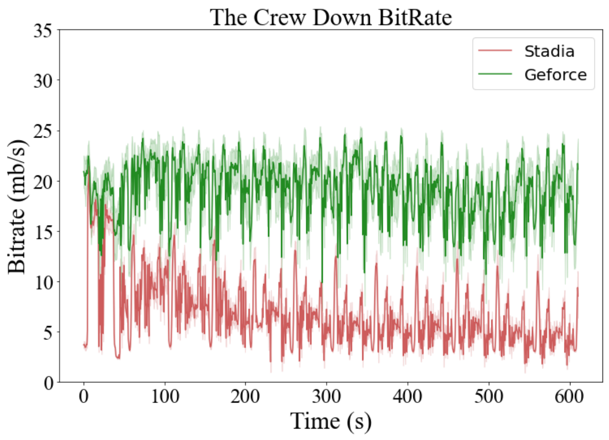
(i) B15 iperf with Luna

Figure 10: Bitrate versus time with different capacity limits.

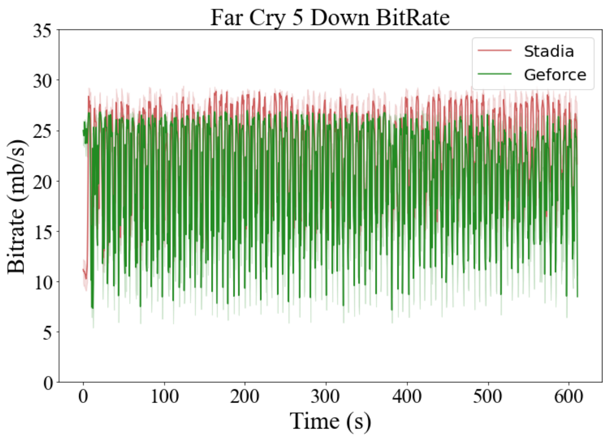
Appendix D BITRATES WITH DIFFERENT GAMES

Cloud systems: Stadia, GeForce
Games: *Ys VIII: Lacrimosa of Dana*, *Far Cry 5*, *Farming simulator*, *Samurai Shodown*
Network: no capacity constraints, no competing flows

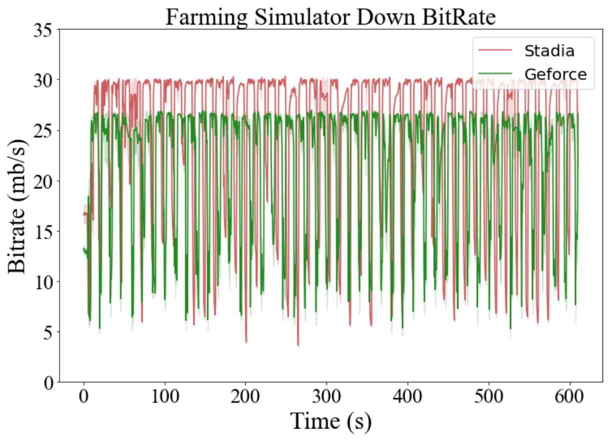
Each game is run for 10 minutes, repeated 20 times, computing means with 95% confidence intervals.



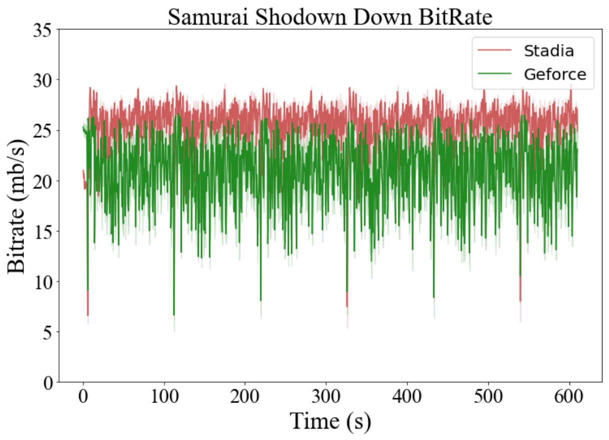
(a) The Crew



(b) Far Cry 5



(c) Farming Simulator



(d) Samurai Shodown