Analysis and QoE Evaluation of Cloud Gaming Service Adaptation Under Different Network Conditions: the Case of NVIDIA GeForce NOW

Mirko Suznjevic, Ivan Slivar, Lea Skorin-Kapov
University of Zagreb Faculty of Electrical Engineering and Computing
Unska 3, Zagreb, Croatia
E-mail: { mirko.suznjevic, ivan.slivar, lea.skorin-kapov}@fer.hr

***Abstract*—Cloud gaming represents a highly interactive service whereby game logic is rendered in the cloud and streamed as a video to end devices. While benefits include the ability to stream high-quality graphics games to practically any end user device, drawbacks include high bandwidth requirements and very low latency. Consequently, a challenge faced by cloud gaming service providers is the design of algorithms for adapting video streaming parameters to meet the end user system and network resource constraints. In this paper, we conduct an analysis of the commercial NVIDIA GeForce NOW game streaming platform adaptation mechanisms in light of variable network conditions. We further conduct an empirical user study involving the GeForce NOW platform to assess player Quality of Experience when such adaptation mechanisms are employed. The results provide insight into limitations of the currently deployed mechanisms, as well as aim to provide input for the proposal of designing future video encoding adaptation strategies.**

Keywords—cloud gaming; adaptation mechanisms; video encoding; Quality of Experience

I. Introduction

The online game industry has recognized the cloud gaming paradigm as a promising shift towards enabling the delivery of high-quality graphics intensive games to nearly any end user device, thus alleviating the need for devices with high-end graphics and processor support. A number of industry leaders have been expanding their services by implementing their own game streaming solutions (e.g., Sony's Playstation Now service[1] and NVIDIA's GeForce NOW (GFN) service[2] as examples of cloud gaming services that allow users online access to a selection of games). Moreover, some game companies provide in-home game streaming that includes the streaming of video games from a local server to other devices in a local network. This approach is applied in Sony's Remote Play service[3], Valve's Steam In-Home streaming service[4] for the PC gaming platform Steam, and NVIDIA GeForce Experience for PCs with selected NVIDIA graphic cards[5].

With powerful servers being responsible for executing the game logic, rendering of the 3D virtual scene, encoding, and streaming game scenes to client devices, the result is a significant increase in downlink bandwidth requirements as compared to "traditional" online games. Thus, a challenge faced by cloud game providers looking to stream their games over the Internet is meeting the Quality of Experience (QoE) requirements of players under various network conditions. A number of previous studies have addressed the relationships between end-user QoE and various network, service, and context factors. While many earlier studies focused on traditional online gaming have provided insight into user-level requirements in terms of factors such as perceived end-to-end latency [1], cloud gaming traffic is inherently different and thus calls for new studies to determine how certain network or application-level factors map to user perceived quality metrics. Researchers have addressed the impacts of latency and/or packet loss on user perceived quality [2], [3], [4], [5], [6], [7], [8], while fewer studies have addressed the impact of different video encoding configurations on QoE [9], [10], [4], [11], [12].

Given that network resources may variate over time (e.g., changes in access network conditions or the number of players accessing a bottleneck link), there is a need for dynamic service adaptation strategies on the game server. Such strategies are based on video codec reconfiguration decisions in terms of parameters such as target bitrate, framerate, and resolution. With the aim of optimizing player QoE, insight is needed with regards to how a given service adaptation strategy impacts QoE. Previous studies [9], [10], [12] have shown that different video adaptation strategies should likely be applied for different games. Both studies [9] and [12] showed that for high bit rates, higher frame rates lead to better overall scores, while for lower bitrates, higher frame rates lead to overall lower scores (attributed to degraded graphics quality in the case of there being more video frames to encode). Furthermore, bitrate reduction was found to have a more significant impact on slow paced games then on fast paced games, while in the case of frame rate the situation was reversed [10].

As pointed out by Hong *et al.* [9], the cloud gaming server has no control over network latency, with packet loss and end-to-end delays resulting in lower effective bandwidth measured by the server. Hence, codec reconfiguration decisions made by the server are driven by measured available effective bandwidth. Following previous studies addressing cloud gaming adaptation strategies, our goal in this paper is to analyze and evaluate the service adaptation logic implemented in the commercial product NVIDIA GeForce NOW, available on the market as of 2015. GFN is a gaming-on-demand service that

[1] https://www.playstation.com/en-ca/explore/playstationnow/

[2] https://shield.nvidia.com/game-streaming-with-geforce-now

[3] https://www.playstation.com/en-ie/explore/ps4/features/remote-play/

[4] http://store.steampowered.com/streaming/

[5] http://www.geforce.com/geforce-experience

connects players to NVIDIAs cloud-gaming supercomputers and enables them to stream PC games to a SHIELD device at up to 1080p resolution and 60 frames per second (fps).

We report on a combination of both objective observations regarding adaptation behavior, as well as subjective user ratings under different network conditions. The contributions of this paper are twofold:

- by objectively characterizing service adaptation behavior under different emulated network conditions, we are able to provide researchers and developers with input for comparing and benchmarking new adaptation strategies against state-of-the-art commercial products, and
- by conducting subjective user tests, we identify potential shortcomings of the currently implemented GFN adaptation logic, which may be used for deriving and optimizing future QoE-driven service adaptation strategies.

Tests have been run in a laboratory testbed with emulated access network conditions, while the game console was connected to GeForce NOW servers via a broadband Internet connection. A number of previous cloud gaming studies have used different platforms such as GamingAnywhere [11], [2], [9], Steam [10], onLive [2], [8], Ubitus [3], or other experimentally set-up platforms. To the best of our knowledge, this is the first paper to report on and empirically evaluate service adaptation mechanisms of the GFN platform. The paper is organized as follows. In Section II we analyse GFN network traffic and service adaptation behavior under different network conditions. Section III reports on the results of a subjective user study conducted to assess the QoE, perceived graphics quality, and perceived game fluidity under different test scenarios expected to trigger service adaptation. Section IV provides concluding remarks and summarizes the implications of the obtained results for the development of future dynamic cloud gaming service adaptation strategies.

II. Analysis of GFN service adaptation behavior

In this section we analyse the network traffic characteristics and service adaptation behavior of the GFN service. The network connection settings which can be manipulated include the characteristics of the incoming video and the target network bandwidth consumption. The characteristics of the incoming video can be tuned to four predefined levels involving the following combinations of resolution and fps: 1080p@60FPS, 720@60FPS, 1080@30FPS, and 720@30FPS. Additionally, there is an *auto* option which allows the GFN service to determine the best combination of fps and resolution to set according to the estimated bandwidth availability. If the *auto* option is not chosen, the user can manually decide whether or not to allow the service to dynamically adapt to network conditions. If the option is not enabled, the incoming stream will be set at a fixed combination of resolution and fps even if bandwidth availability is severely reduced. The Shield console has the option of outputting a 4K resolution video to the TV to which it is connected, but currently this option is only reserved for NVIDIA GeForce Experience (i.e., streaming games from a local PC), but not for GFN. The network bandwidth consumption can also be set using the *auto* option or can be manually set to any value between 4 Mbit/s to 30 Mbit/s. Although the suggested values of bandwidth consumption on the GFN web page have been listed as 10, 20 and 30 Mbit/s, the game stream can be delivered even at 4 Mbit/s, although with significantly reduced video stream quality (usually with 540p@30FPS).

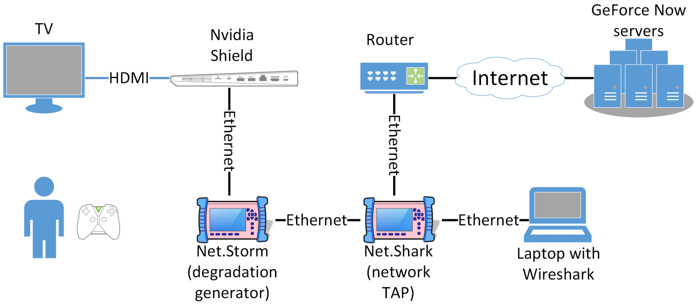

Fig. 1: Laboratory testbed

TABLE I: Measured parameters of our network compared to required and recommended parameters for GFN

	Estimated	Required	Reccomended
Bandwidth	> 50 Mbit/s	> 20 Mbit/s	> 50 Mbit/s
Frame loss	$< 0\%$	$< 3\%$	$< 1\%$
Jitter	18 ms	< 80 ms	< 40 ms
Latency	22 ms	< 80 ms	< 40 ms

Our laboratory testbed is shown in Figure 1. A player uses a wireless gamepad for controlling the game. The shield console is connected via an HDMI cable to the television set on which the game content is displayed. Shield is connected to the Internet and GFN servers via Albedo's Net.Storm[6] and Net.Shark[7] devices. Net.Storm is a commercial grade network emulation device that can apply a wide range of network impairments to IP/Ethernet streams, including bandwidth limitations, latency, and loss via a variety of modes (e.g., bursts of loss or exponentially distributed latency). Net.Shark is a portable network tap which was used to aggregate and replicate the traffic passing between the Shield console and GFN servers. Traffic was then sent to a laptop and captured using Wireshark. In this way we eliminate the impact of packet capture on the processing power of the end device.

Prior to initiating game play, the Shield console has a network test option in which the characteristics of the network are estimated. We note that under unimpaired conditions, our network has been graded as "Excellent network". The values of evaluated, required, and recommended network parameters for GFN are listed in Table I (parameters are depicted as reported by the Shield console). Video stream parameters were measured through a built-in tool in the Shield console. When activated, data in the following format was dynamically portrayed in the upper right corner of the screen: *<resolution>@<frame rate> <bandwidth used> <percentage of available bandwidth used> <number of lost frames>*.

A. Traffic analysis

To obtain insight into the traffic characteristics of the GFN service, we recorded and analysed GFN traffic for three different games, while setting the *auto* adaptation option and with no degradations imposed in the network. The following three games were tested: *Dirt 3* as an example of a racing

[6] http://www.network-testers.com/albedo_net_storm.html

[7] http://www.albedotelecom.com/pages/fieldtools/src/netshark.php

game, *Ultra Street Fighter IV* as a 2D fighting game, and *Pumped BMX +* as an arcade sports platform game. The same games were subsequently used in our subjective studies, reported in the following section. We captured the traffic of approximatively 30 seconds of gameplay for each game, resulting in approximately 390 MB of traffic. Traffic analysis was done using the tools *OmniPeek* by WildPackets, and *Wireshark*. We shortly list the traffic characteristics.

GFN uses RTP over UDP to deliver video content, which in our measurements was always delivered from a single IP address. Figure 2 illustrates the bandwidth usage of all three tested games. The bandwidth usage greatly depends on the characteristics of the video being sent. Consequently, the greatest variation may be observed in the case of the BMX game, where game play levels are short and there are stationary points in the video when levels are reset, while in Dirt 3 there is almost no variation as the state of the virtual world is relatively constant, corresponding to car racing. Traffic is very asymmetric, with the majority of packets and data being sent in the downlink direction (95.45%). The majority of downlink packets is fixed at 1080 bytes (B) (over 90%), while the remaining packets are mostly smaller then 126 B. The distribution of packet sizes in the uplink direction has discrete steps with prominent values (102 B, 118 B, 142 B, and 150 B).

B. Adaptation to network delay, delay variation, packet loss, and bandwidth shaping

In the effort to better understand the adaptation algorithm employed by the GFN service, we introduced different amounts of bandwidth limitations, latency, delay variation, and packet loss onto the network link using the Net.Storm emulation device. All tests have been performed multiple times to ensure validity of observed behavior. We note that prior to running all tests, the Shield console network test was run on an unimpaired network to evaluate network conditions. Once the network test is performed, it appears that the service remembers the conditions in which the network test has been last performed. For example, if throughput is limited to 10 Mbit/s prior to running the network test, the service will not try to push more then 10 Mbit/s at any time, even after the bandwidth restriction has been lifted.

Latency. Our goal was to test GFN service behavior in light of a small amount of latency dynamically added during game play. Surprisingly, when inserting an additional latency of 10 ms or more (we tested adding delay of 100, 50, 20, and 15 ms) in the downlink direction *during* game play, we observed that bandwidth consumption quickly drops to approximately 2 Mbit/s, and within seconds the streamed video drops to the lowest possible setting (in this case 30FPS@540p). However, if the latency is introduced before the game itself was started, this degradation does not occur. This unexpected behavior is either a weakness of the system in terms of bandwidth estimation algorithm, or that the specific way in which the Net.Storm emulator adds latency somehow "tricks" the system. To rule out the second case, we conducted the same tests using a different emulator, namely the freely available IMUNES emulator/simulator tool[8], and results in terms of GFN service behavior proved to be the same. This leads to two conclusions regarding the GFN adaptation algorithm: **the bandwidth estimation and adaptation algorithm is somehow based on RTT**, and **bandwidth adaptation is only triggered during gameplay and not in the game selection screen**. When latency added during game play is removed, the system recovers to nominal settings (1080p@60FPS with bandwidth usage around 30 Mbit/s) within seconds. If the added latency is not removed, the system again recovers, but much slower and differently depending on characteristics of the video stream. We tested two scenarios in Dirt 3: 1) with active gameplay - the player continues to drive the car, and 2) with passive gameplay - the car is stopped and there is no action. The results are depicted in Figure 3 and Figure 4. First, we can conclude in both cases that the adaptation algorithm values more frame rate then resolution, as reductions are first observed in terms of resolution, followed by frame rate. Moreover, in the recovery phase, the frame rate is increased first, and afterwards the resolution. In the active game play mode, it took approximately 2 minutes for the game to recover, and the recovery started only when the GFN service reported that the value of the *percentage of available bandwidth used* reached 0%. Currently we do not have an explanation why the recovery starts at 0% of available bandwidth used. On the other hand, in the passive game play mode, the resolution and frame rate increased significantly prior to the bandwidth. We attribute this to the complexity of the video, as the car was stopped and the image was relatively static, enabling the resolution and frame rate to reach peak values even for 2 Mbit/s. Based on this we conclude that **the adaptation of fps and resolution is separate from the bandwidth evaluation algorithm and is likely based on spatial and temporal video complexity and the current bitrate the video coder has available**. For passive recovery we see that the system first recovers to around 14 Mbit/s which is sufficient for full quality of the still image, and increases to 30 Mbit/s immediately after gameplay is continued. Also we assume that **the GFN bandwidth estimation algorithm runs in very small time periods or possibly even at the level of several or single video frames**. Adding 10 ms of latency stops some of the packets from arriving for that period. In this initial halt of packets, the system recognizes that it is not receiving enough data and responds by reducing the amount of data sent. The question remains as to why it takes so long for the system to recover while this added latency is present, while the system responds almost immediately if the added latency is removed.

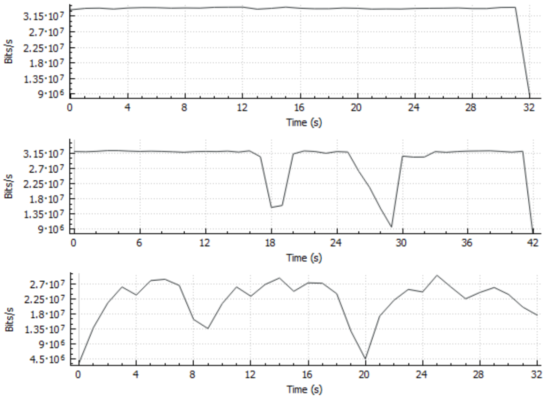

Fig. 2: Bandwidth usage for (top to bottom) Drift 3, Ultra Street Fighter 4 and Pumped BMX +

[8] http://imunes.net/

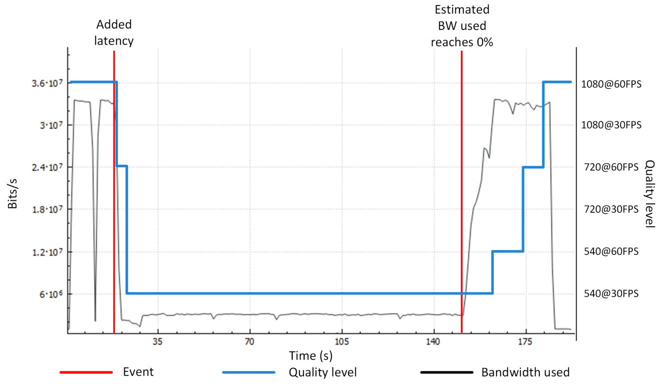

Fig. 3: Adaptation with active gameplay in Dirt 3

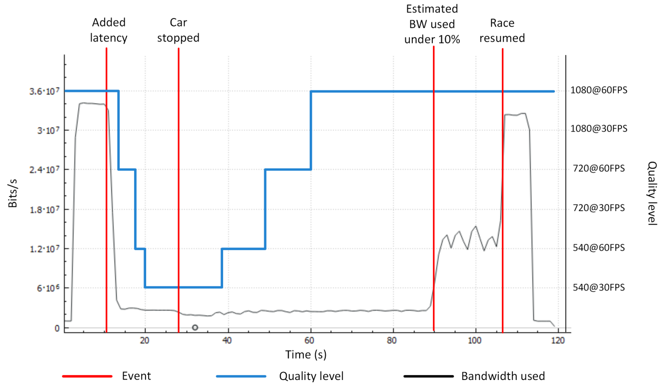

Fig. 4: Adaptation with passive gameplay in Dirt 3

Delay variation (jitter). We tested the impact of adding delay variation by inserting latency according to a uniform distribution ranging from 10 ms to 45 ms. It should be noted that delay is inserted per each single packet, and that reordering of packets was allowed. In this way inserted delay of one packet does not influence the subsequent packet. This approach does not significantly change the general statistics of inter-arrival times of subsequent packets on the receiver side (while it does change ordering of packets). The observed results were time of two subsequent packets. The system immediately reduced the amount of data sent to only 2 Mbit/s and dropped the resolution and frame rate to the minimal supported value (540p@30FPS). The difference with respect to inserting deterministic latency was that the system did not recover to the full (peak) quality.

Packet Loss. Tests showed that the GFN service is very resilient to packet loss, likely due to the use of Forward Error Correction mechanisms. Even at loss rates of 10%, the gameplay was fluid and only minor glitches occurred. On the other hand, there was no reaction of the bandwidth adaptation algorithm, which remained at a constant rate even with losses of 10% in both uplink and downlink directions. This leads us to the conclusion that **the bandwidth estimation algorithm is impacted primarily by latency when detecting possible network congestion**.

Bandwidth limitation. We tried limiting the bandwidth with two techniques available on the Net.Storm device: *policing* and *shaping*. Both techniques are based on a token bucket system where in the case of shaping, packets are put into a queue if tokens are spent, and in case of policing the packets are immediately dropped if there are no tokens in the bucket. Because the system reacts by reducing bandwidth consumption only when latency is added, we chose to limit the bandwidth with the shaping option (in the case of using the policing option, the service degraded severely and eventually disconnected). The system proved quite responsive and limited the bandwidth sending rate within seconds. We observed what combinations of resolution and frame rate occur for different bandwidth limitations and results are depicted in Figure 5. For some bandwidth values two different combinations of resolution and frame rate were noticed depending on the characteristics of the video (e.g., in Drift 3 a drop to lower settings would often occur when the car would crash off the road). From Figure 5 it is noticeable that for Pumped BMX + much lower bandwidth is required to reach maximal quality level than in other two games (13 Mbit/s as opposed to 19 Mbit/s), while other two games quite similarly adapt to bandwidth limitations. **These adaptations are based on spatial and temporal video characteristics for each game** - Pumped BMX + has significantly lower graphics detail and is less dynamic then other two tested games.

Finally, all tests were conducted using all three tested games, and no significant difference in behavior was detected. These leads to the conclusion that **the currently implemented GFN adaptation algorithms are not dependent on particular game nor game type being played, but only on video characteristics**.

III. QoE EVALUATION

A. *Methodology*

To evaluate the impact of the GFN service adaptation algorithm on player QoE under various network conditions, we conducted a user study consisting of players taking part in approximately 45 minute long gaming sessions that were run in the previously described lab testbed (the network TAP device was removed to eliminate the possibility of additional delay induced by this device). The three previously mentioned games were used for testing. Although all three games belong to different game genres, we acknowledge the fact that they all fall under the category of more dynamic games in terms of gameplay pace, but due to a limited selection of slow-paced games with a short learning curve provided by the GFN service, we opted for these games as they are significantly different in terms of game characteristics (e.g., camera perspective, and level of graphics detail). All games were played by using auto settings for video resolution and frame rate (i.e., under unimpaired conditions the quality was 1080p@60FPS),

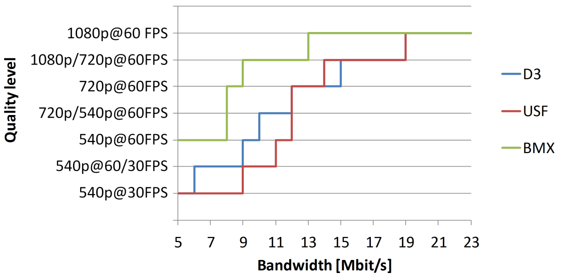

Fig. 5: Quality levels of tested games on various bandwidths

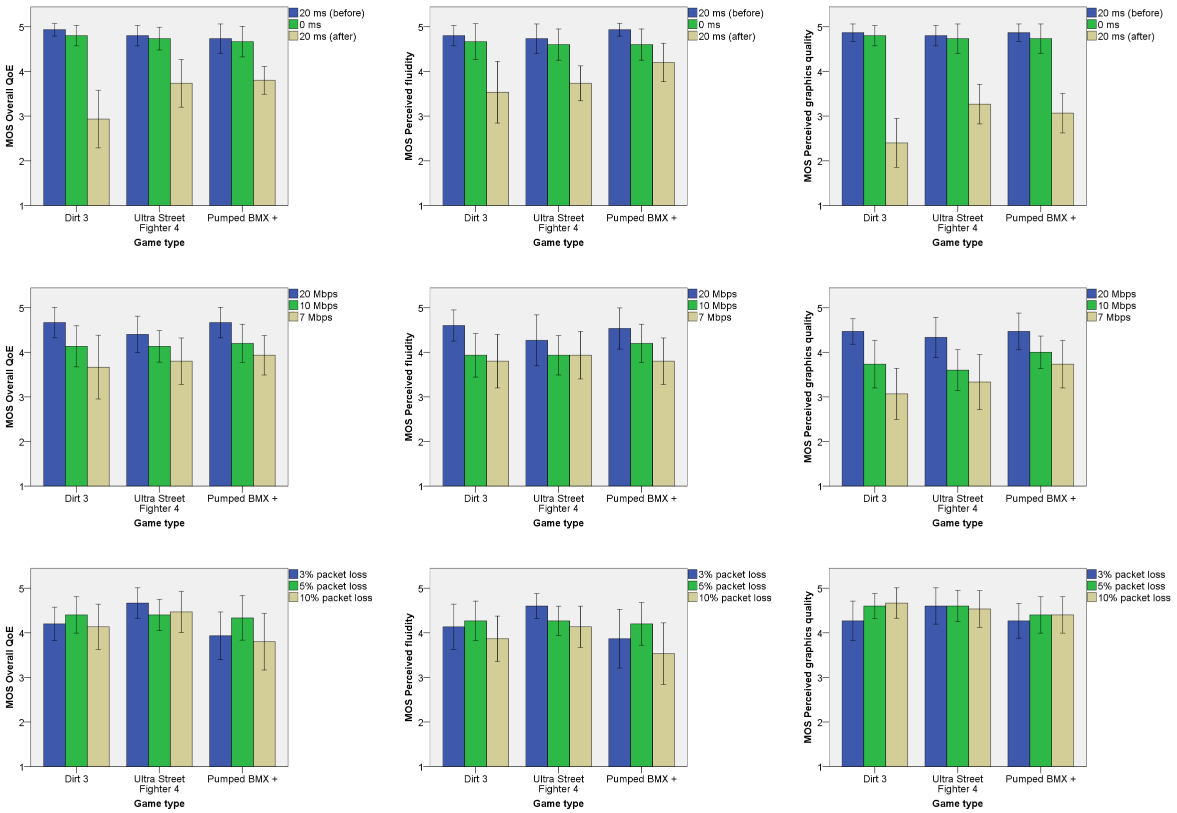

Fig. 6: Subjective scores for QoE and its features

and auto settings for the network bandwidth (at the time of the study, the maximum bandwidth that the GFN service used was 30 Mbps). The participants were 15 adults (14 male and one female), aged between 22 and 33 (average age 25.93, median age 26). All participants were self-reported as highly experienced players. To invoke adaptation, we manipulated the parameters latency, packet loss and bandwidth. Our aim was to investigate how users rate overall QoE, perceived graphics quality, and perceived fluidity after service adaptation is invoked due to changing conditions. In accordance with the GFN service requirements and recommendations, we used three levels of packet loss (3%, 5% and 10%), and three levels of available bandwidth (20 mbps, 10 mbps and 7 mbps). Regarding latency, our goal was **not** to test the impact of different latency values, but rather to quantify the impact of the observed phenomena previously described corresponding to inserting additional latency into an already initiated gaming session. This was accomplished by testing three scenarios: 20 ms added **prior** to game play (denoted on results graphs as *20 ms (before)*); no latency; and the addition of 20 ms latency **during** game play (denoted on results graphs as *20 ms (after)*).

Considering the 3 test scenarios for each of 3 parameters, tested across 3 games, the study included a total of 27 test scenarios. Test scenarios were tested by each participant, according to a randomized sequence (per parameter) to avoid possible bias and ordering effects (i.e., test scenarios corresponding to a certain parameter manipulation were grouped together). Only in the case of latency testing, scenarios were kept in the same order, adding 20 ms latency prior to game play, removing the latency, and then reintroducing the latency during game play. Before each of the gaming sessions, the participants were given a small amount of time to get acquainted with the tested games and their controls. After finishing each test scenario (which lasted between 30 seconds and 1 minute, depending the game), the participants were instructed to fill out a questionnaire and report overall QoE, perceived fluidity, and graphics quality (all reported on a 5-point ACR scale). Additionally, players expressed their willingness to continue playing under the current test conditions.

B. *Results*

The average subjective scores for QoE and its features under various network conditions are shown in Figure 6. Concerning latency fluctuations during gameplay, results show that in the test cases when latency was reintroduced into the system after an already initiated gaming session, the average scores for overall QoE and its observed features are significantly lower than in test cases without artificial latency. This is particularly visible for the averages scores of graphics quality for Dirt 3, that are significantly lower in comparison with the other two games. Although Dirt 3 is a highly fast-paced game, the level of detail and overall graphics quality of the game is high enough that players can notice lower video resolution and frame rate values (i.e., video bitrate of 2 Mbps, and video quality of 540p@30FPS) that occurred as a result of service adaptation. Given that the added latency of 20 ms is very low and falls within the specified GFN requirements, it is clear that the corresponding QoE degradation is not a direct result of the latency, but rather the result of currently implemented GFN bandwidth estimation and service adaptation mechanisms.

With respect to service adaptation due to increased packet loss, it is evident that even though the subjective scores are in general lower than in test scenarios when the service is running under perfect conditions, user ratings confirm that the GFN service is very resilient to packet loss, without having to employ any service adaptation techniques. The impact of service adaptation strategies on users' QoE due to different amounts of allocated bandwidth showed similar trends in the QoE assessment as in a case of the latency scenarios. The MOS scores for overall QoE decreased, likely due to lower graphics quality, while mean values of fluidity scores remained relatively high (MOS score around 4) for all tested games (we note that the game never reduced the frame rate below 30FPS). While reducing the bandwidth to 7 Mbps, which is below the minimum required by the GFN service, resulted with lower average QoE scores, it should be noted that for Pumped BMX, the MOS score was slightly below 4 which is quite high. This indicates the potential for overall bandwidth optimization strategies based on game characteristics.

Finally, Figure 7 portrays the willingness of players to keep playing under certain test conditions. The results clearly show that in the case of minimum added latency during game play, a significant percentage of players would opt to end game play, again confirming the potentially significant impact of bandwidth estimation and corresponding service adaptation algorithms on QoE, rather than the direct impacts of latency itself. Furthermore, results show that at bandwidth limitations of 7 Mpbs, in total a significant portion of ratings showed players not willing to keep playing. However, when considering this issue on a per-game level, the per-game QoE scores indicate that for certain game types, such bandwidth limitations may be considered acceptable.

IV. Conclusions and future work

In this paper we have evaluated the bandwidth adaptation strategy of NVIDIA's GFN cloud gaming service based on a combination of both objective observations regarding adaptation behavior, as well as subjective user ratings under different network conditions. The reported observations may provide useful input for researchers and developers in terms of comparing and benchmarking cloud gaming adaptation strategies. Based on obtained results, and building on findings reported in previous work, we draw the following conclusions. The GFN video codec sending rate is adjusted based on latency and bandwidth limitations, but not by packet loss. Given that service adaptation strategies are driven by client-side bandwidth estimation algorithms, inaccurate estimations may result in severe QoE degradations due to the suboptimal configuration of video codec parameters.

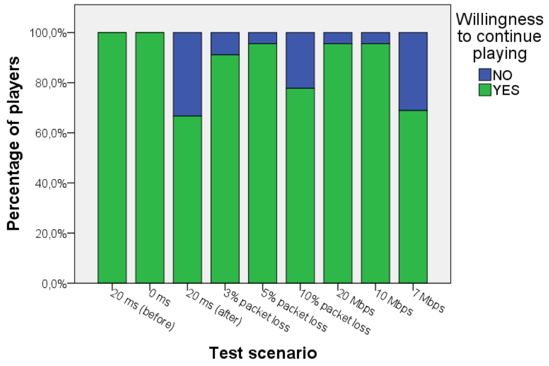

Fig. 7: Willingness to continue playing

Furthermore, despite the fact that a number of studies addressing cloud gaming QoE have recognized game genre as a key context QoE influence factor, today's state-of-the-art commercial solutions such as GFN are still not taking into account game genre while performing dynamic service adaptation due to resource availability constraints. One obstacle to performing game genre-aware adaptation is the lack of an existing classification of digital games based on objective game characteristics that could be used to categorize games for the purpose of assigning appropriate and QoE-driven adaptation strategies. Our future studies will focus on specifying QoE-driven video encoding adaptation strategies for different available system and network conditions, with the goal of exploiting such strategies for optimizing cloud gaming QoE.

V. Acknowledgements

This work has been supported by the Croatian Science Foundation under the project UIP-2014-09-5605 (Q-MANIC).

References

[1] M. Claypool and K. Claypool, "Latency and player actions in online games," *Commun. ACM*, vol. 49, no. 11, pp. 40–45, Nov. 2006. [Online]. Available: http://doi.acm.org/10.1145/1167838.1167860

[2] M. Claypool and D. Finkel, "The effects of latency on player performance in cloud-based games," in *Proc. of 13th Annual IEEE/ACM NetGames*, 2014, pp. 1–6.

[3] Z.-Y. Wen and H.-F. Hsiao, "QoE-driven performance analysis of cloud gaming services," in *Multimedia Signal Processing (MMSP), 2014 IEEE 16th International Workshop on*. IEEE, 2014, pp. 1–6.

[4] Y. Liu, S. Wang, and S. Dey, "Content-aware modeling and enhancing user experience in cloud mobile rendering and streaming." *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 4, no. 1, pp. 43–56, 2014.

[5] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "Gaming in the clouds: QoE and the users' perspective," *Mathematical and Computer Modelling*, vol. 57, no. 11, pp. 2883–2894, 2013.

[6] V. Clincy and B. Wilgor, "Subjective evaluation of latency and packet loss in a cloud-based game," in *Information Technology: New Generations (ITNG), 2013 Tenth International Conference on*. IEEE, 2013, pp. 473–476.

[7] S. Möller, D. Pommer, J. Beyer, and J. Rake-Revelant, "Factors Influencing Gaming QoE: Lessons Learned from the Evaluation of Cloud Gaming Services," in *Proceedings of the 4th International Workshop on Perceptual Quality of Systems (PQS 2013)*, Sep. 2013, pp. 1–5.

[8] Y.-T. Lee, K.-T. Chen, H.-I. Su, and C.-L. Lei, "Are all games equally cloud-gaming-friendly? An electromyographic approach," in *Network and Systems Support for Games (NetGames), 2012 11th Annual Workshop on*. IEEE, 2012, pp. 1–6.

[9] H. Hong, C. Hsu, T. Tsai, C. Huang, K. Chen, and C. Hsu, "Enabling adaptive cloud gaming in an open-source cloud gaming platform," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–14, 2015.

[10] I. Slivar, M. Suznjevic, and L. Skorin-Kapov, "The impact of video encoding parameters and game type on QoE for cloud gaming: A case study using the Steam platform," in *Quality of Multimedia Experience (QoMEX), 2015 7th Intl. Workshop on*, May 2015, pp. 1–6.

[11] J. Beyer, R. Varbelow, J.-N. Antons, and S. Möller, "Using electroencephalography and subjective self-assessment to measure the influence of quality variations in cloud gaming," in *Quality of Multimedia Experience (QoMEX), 2015 Seventh International Workshop on*, May 2015, pp. 1–6.

[12] I. Slivar, L. Skorin-Kapov, and M. Suznjevic, “Cloud Gaming QoE Models for Deriving Video Coding Adaptation Strategies,” in *To appear in Proc. of ACM Multimedia Systems 2016 (MMSys’16)*. ACM, 2016, pp. 1–12.