Prediction of Video Quality Degradation on a Cloud Gaming Platform

Alex Amaral
Universidad de la República
Montevideo, Uruguay
alexdaniel.amaral@gmail.com

Alejandra Armendariz
Universidad de la República
Montevideo, Uruguay
aarmendariz@fing.edu.uy

Santiago Erramuspe
Universidad de la República
Montevideo, Uruguay
santiago.erramuspe@fing.edu.uy

Jose Joskowicz
Universidad de la República
Montevideo, Uruguay
josej@fing.edu.uy

Mengying Liu
China Mobile Research Institute
Beijing, China
liumengying@chinamobile.com

***Abstract*—Cloud gaming has become a promising way to play high-quality video games on relatively inexpensive devices. Cloud gaming platforms run interactive video games remotely in the cloud and stream multimedia to the gamer's device via a telecommunications network. These platforms have many benefits, but also pose several problems and challenges, as they are highly dependent on the network from servers to users. This work aims to study the effect of network impairments on video quality, and to predict video quality degradation in real time. In particular, it was determined that latency is a good predictor of packet loss, which was found to result in visual degradation. Based on this, an algorithm was designed to predict real-time visual degradations before they actually occur.**

***Index Terms*—Multimedia Transmission, Multimedia Service, Cloud Gaming, Visual Degradations, Real-time Prediction, Quality of Experience**

I. Introduction

The world of video gaming on personal devices has become very popular over the past 30 years. On traditional systems, players have to install video games on their physical machines in order to play them. Cloud gaming (CG), on the other hand, uses a client-server architecture where the games are uploaded and processed in the cloud and the user sees the video of the game being played on a thin client [1]. CG provides easy access to games without the need for an expensive gaming computer [2].

However, while broadcast of live events appears to be a mature technology and delays are acceptable, the interactivity of network video games makes a CG system harder to optimize than a classical multimedia streaming system [3]. The purpose of this work is to identify and measure parameters in the network that affect audio and video transmission, and thus develop an automatic system to anticipate degradations.

This document describes the early detection of potential issues in the network that may affect the playability of video games using a CG platform. In the following sections, we will show how the designed algorithm works to accomplish the task of predicting visual degradations. In addition, the different scenarios in which different types of video games have been tested are shown, along with the results.

II. Cloud Gaming Issues and Possible Solutions

CG platforms rely heavily on the network connecting the server to the client. Poor network performance can have a significant impact on user Quality of Experience (QoE). If latency is high, gameplay may suffer, especially in games that require fast response. In addition, providing high-quality graphics requires high bandwidth [4]. Figure 1 shows a snapshot from the video game "Alien Rage", in which the visual degradation is caused by the problems mentioned.

Fig. 1. Screenshots of the game Alien Rage with and without visual degradations (top and bottom respectively).

Servers closer to users, edge computing, deployment of the 5G cellular network, and use of Wi-Fi networks in the 5 GHz band will all help to minimize network problems [5]. Nevertheless, these networks are not universally deployed nowadays, and cloud gamers suffer degradations due to network issues. Therefore, network-independent solutions to improve the gaming experience must be implemented, through optimization tools installed on the client and/or server, using systems that predict possible degradations during gameplay.

III. Prediction of video quality degradation

A. Measurement of network parameters

This work was developed using the cloud platform powered by ABYA [6], which provides CG services in Latin America. For this research, ABYA team created a test bed, including a real-time logging system. While a game is being played, the measurements of a number of chosen network variables are logged in real time on the client side. These variables include latency, video frame losses, estimated bandwidth, network packets out of order, among others. The log file generated on the client side can be accessed at the end of the game and also in real time during the game.

ABYA's platform uses 60 fps for the video stream, so a frame is received every 16.7 ms. The logs are updated every 20 ms and the different network parameters are calculated for each 20 ms window. As an example, some of the variables are defined as follows:

- *Average Network Latency (AvgNetLat):* This variable measures the delay that packets experience from the time they are sent from the server until they are received by the client. It can be calculated by dividing the total delay of all the received packets by the number of packets received during the measurement interval, as shown in Equation 1:

$$AvgNetLat = \frac{\sum(T_{arrival} - T_{departure})}{Packets\ received} \quad (1)$$

where $T_{arrival}$ and $T_{departure}$ represent the time when a packet is received and sent, respectively.

- *Estimated bandwidth (estBW)*: This metric can be calculated as the sum of all received bits divided by the duration of the measurement window, as shown in Equation 2. It is important to note that this estimation does not reflect the actual capacity of the communication channel, but rather the bandwidth of the audio and video streams received by the client.

$$estBW = \frac{\sum Bits\ received}{Window\ duration} \quad (2)$$

B. Test bed

Initial tests used a typical home Wi-Fi network to determine the most common degradations perceived by users on such networks. Based on the results obtained, a controlled test bed was developed and deployed.

Our test system consists of a computer running the CG client application and other programs and tools used in the project. It connects to the Internet through another computer configured as a Layer 2 bridge running a network emulator. The connection setup diagram is shown in Figure 2. The client is directly connected to the bridge using an Unshielded Twisted Pair (UTP) cable, as is the connection from the bridge to the Internet router. The Internet router was used solely for this test, with no other source of Internet traffic. With this setup, the test bed has full control of the wired network that simulates the user's environment. The ABYA server used for this test, located at the cloud, was not loaded with other users, so no other sources of degradations or impairments were present during the tests.

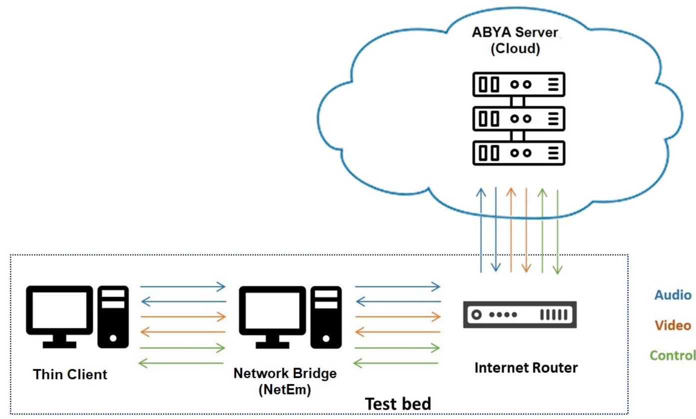

Fig. 2. Test bed connection diagram

To implement the bridge, we used an Ubuntu 20.04 computer with two LAN network interfaces. The *brctl* command from the *bridge-utils* package is used to bridge network segments connected to both interfaces and allowing the exchange of IP packets transparently between the client and the CG server that is in the cloud.

Emulation of network conditions is done by the NetEm application, which also runs on the computer that acts as a network bridge. After conducting several preliminary tests, it was decided that the channel bandwidth limitation is the most critical parameter to consider, emulating what happens in an overloaded home Wi-Fi network. A restricted channel bandwidth results in an increase in the delay of outgoing packets in the downlink stream and leads to packet loss, which produces video frame loss, which in turn is perceived as visual impairments.

Four representative typical situations encountered in domestic Wi-Fi contexts were used to test the impact on different games. Figure 3 presents the selected test settings, each with a duration of 180 seconds. To systematize the implementation of the conditions to be emulated, a script was developed. This script automatically makes the changes in the NetEm, according to each test setting.

C. Video frame loss prediction

Video degradations occur when a video frame is lost. By analyzing the logs and various network metrics, it was possible

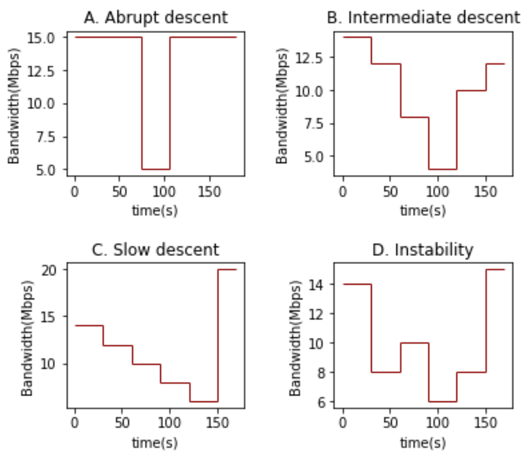

Fig. 3. Test settings of bandwidth limitation implemented with NetEm

to establish a strong correlation between a frame loss and the network latency, with latency increasing sharply just before the loss occurred in all cases. Figure 4 shows a typical case. At the bottom of the figure the estBW is shown, in the middle of the figure the AvgNetLat is presented, and at the top of the figure the number of lost frames. It can be seen that, when the bandwidth decreases below a certain limit, the packets' latency suddenly increases, and a few instants later, several frames are lost.

Based on these facts, we developed a loss predictor program to detect degradations in real-time. The program was implemented in Python using Jupyter Notebook by Anaconda Navigator. During gameplay, the program runs in parallel in the client's computer, it checks the logs in real time, and warns in advance of any impending degradation. The program implements moving windows to measure the accumulated latency and to estimate the bandwidth. We chose to use windows with summatory of latency values to prevent false positives alarms of degradation. According to our tests, individual values with high latency or outliers do not indicate that quality will decrease. In contrast, high latency accumulation in the form of "ramps", as shown in Figure 4, was a clear indication of quality degradation.

The windows used in this system have a fixed length and follow a First in, First out (FIFO) queue structure. At regular time intervals, the window samples various parameters and records the data sequence, updating the accumulated latency and received bytes as the game session progresses. An algorithm analyzes this data and triggers an alarm when the calculated values exceed a certain threshold, indicating an imminent degradation.

To produce stable and reliable results, a window with a length of 500 ms is used by summing the latency values and averaging the bandwidth every 20 ms. The threshold value for the integrated latency was optimized by playing different games under various network conditions. When the program detects that the sum of the latency exceeds the threshold, it issues a degradation alarm. The results obtained are presented in the next section.

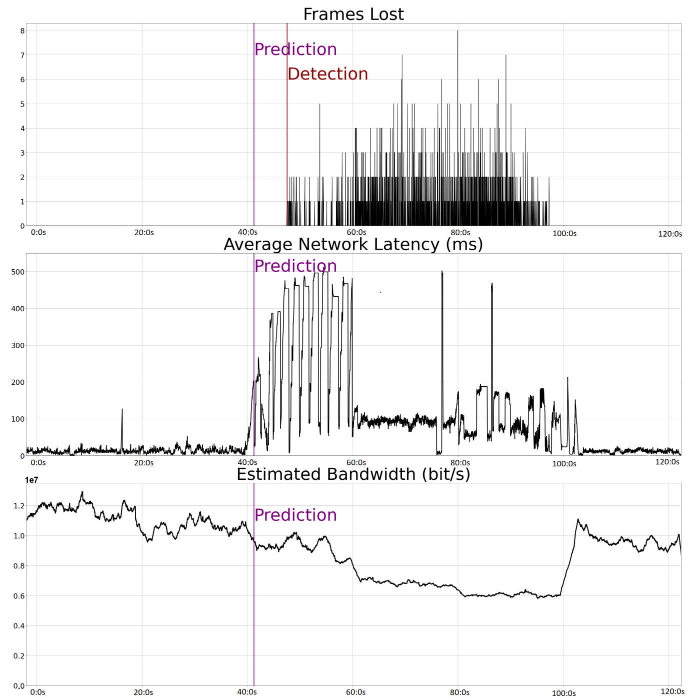

Fig. 4. Top: frames lost, degradation prediction and detection. Center: network latency. Bottom: estimated bandwidth received by the client

IV. Results

The algorithm was tested on 15 video games, with a total of 183 games played. Table I lists all games used, with their respective genres, brief descriptions, and game title abbreviations. Fast-paced games were used in order to test the algorithm with games that require a good connection. Genres tested include role-playing games (RPG), first-person shooters (FPS), racing games, and fighting games. These genres are large families with many sub-genres and their combinations. A brief description of the games is included in Table I to supplement the information on each of them. Real-time strategy (RTS) genre was not tested. Games of this genre have lower bandwidth requirements and lower user responsiveness requirements. Therefore, it is believed that the predictions that worked for RPG and FPS, which have higher responsiveness requirements, will also work for RTS games.

Table II reports the results obtained from the playthroughs made for the 15 titles in the network settings presented in Section III. At this point, it is worth analyzing a specific playthrough made for a better understanding of the data collected.

Figure 5 presents data from a playthrough of the TAN game, which was conducted under the abrupt descent network setting discussed in Section II. The limited bandwidth associated with this setting is depicted in red in Figure 5 labeled as NetEm. As explained in Section III, visual degradations are correlated with consecutive frame losses. Based on the information

TABLE I
LIST OF TESTED GAMES

Game Title	Genre	Brief description	Abbr.
Alien Rage	FPS	FPS game set in space with aliens	AR
Cyber Hook	FPS	First person 3D parkour skills game	CH
Classic Racers	Racing	Racing game with classical cars	CR
Extinction	RPG	Action RPG sword fighting game	EXT
Gravel	Racing	Extreme racing game	GRA
Grip	Racing	Futuristic combat racing game	GRI
King of Fighters XIII	Fighting	Fighting game in The King of Fighters series	KOF
Lord of the Fallen	RPG	Action RPG game set in a medieval fantasy world	LOTF
MXGP 2020	Racing	Official Motocross World Championship game	MXGP
Outcast: Second Contact	RPG	RPG action adventure set in an alien world	OSC
Stay Safe	Racing	Racing game with obstacles die and retry style	SS
Styx: Master of Shadows	RPG	RPG game set in a dark fantasy universe	STYX
Tandem	RPG	Puzzle platformer game	TAN
V-Rally 4	Racing	Realistic rally racing game	VR4
WRC 8	Racing	Official World Rally Championship game	WRC

TABLE II
RESULTS OBTAINED WITH FIFTEEN DIFFERENT GAMES TESTED WITH FOUR DIFFERENT NETWORK CONDITIONS

Title	Launched games	Total visual degradations	Predictions	Tested network conditions
AR	7	11	12	A - D
CH	20	32	37	A - B - C - D
CR	20	37	39	A - B - C - D
EXT	7	8	11	A - B
GRA	7	7	8	A - B
GRI	20	29	32	A - B - C - D
KOF	20	39	43	A - B - C - D
LOTF	20	25	25	A - B - C - D
MXGP	7	10	10	A - D
OSC	7	14	18	B - D
SS	7	8	9	A - B
STYX	7	10	13	A - C
TAN	7	17	20	A - B
VR4	7	7	7	A - B
WRC	20	15	16	A - B - C - D
TOTAL	183	269	300	

shown in the graph, it can be inferred that two instances of visual degradation occurred during the playthrough.

As discussed in Section III, whenever the accumulated latency in a moving time window exceeds a certain threshold, the algorithm performs a prediction of consecutive frame losses. The graph titled "average network latency" at the middle of Figure 5 confirms this behavior by showing the two predictions made during the playthrough. Table II contrasts the predictions made with the visual degradations that actually occurred.

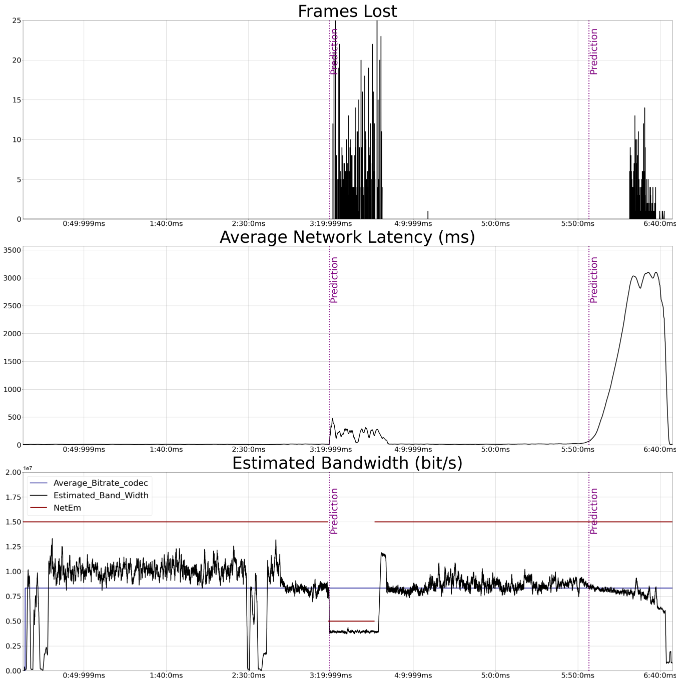

Fig. 5. Top: frames lost, degradation prediction and degradation detection. Center: network latency. Bottom: estimated bandwidth received by the client, NetEm configuration and bitrate average codec configuration

TABLE III
SUMMARY RESULTS

	Total number of predictions	Successful predictions	Unpredicted degradations	False Positives
Quantity	300	269	0	31
Percentage	100%	89.7%	0%	10.3%

Finally, the results are summarized in Table III and expressed as percentages. Out of a total of 183 games played with 15 different titles and 4 network scenarios, the algorithm made 300 predictions out of a total of 269 observed visual degradations. This corresponds to 0% of unpredicted degradations and a false positive prediction rate of 10.3%.

It is important to note that all visual degradations were successfully predicted. In this application, it is better to have a small amount of false positives, than to miss some real degradation event [7].

V. NEXT STEPS AND FUTURE WORK

The false positive results are being analyzed in order to minimize them without losing real events. We are also experimenting with automatically reducing the video codec bitrate after a degradation has been predicted. This should maintain good gameplay at the expense of some reduction in video quality [8]. We will investigate how much the codec bitrate should be reduced in order to eliminate frame loss and minimize the impact on quality. Once an automatic bitrate reduction is performed after a frame loss prediction, the next challenge will be to automatically increase the bandwidth of the codec once the network issues are resolved.

All the developed algorithms can then be implemented in the Cloud Platform components (server and client), thus improving the game playability.

VI. Conclusion

Cloud gaming platforms have emerged as a promising means to play high-quality video games on low-cost devices. However, this also poses several problems and challenges as it is highly dependent on the network bandwidth, from servers to users.

In this work, it was detected that the network latency can be used as a good predictor of video frame losses, which are the cause of video degradation. In this context, a real time algorithm was developed to predict future visual degradations. This algorithm runs at the client's PC, and can be easily incorporated to the cloud gaming application.

The developed algorithm was tested with 15 games of different genres, over 4 different network conditions, resulting in over 180 played games. It was verified that the algorithm predicts 100% of the video degradations events, although it also has 10.3% false positives.

VII. Acknowledgements

The authors would like to thank the ABYA team for their support and contribution to this work.

References

[1] W. Cai et al., ”A Survey on Cloud Gaming: Future of Computer Games”, in IEEE Access, vol. 4, pp. 7605-7620, 2016

[2] P. Shrivastava and M. Damle, “Investment decision in cloud gaming-based businesses opportunities: An analysis of the cloud gaming industry,” 2022 International Conference on Decision Aid Sciences and Applications (DASA), 2022, pp. 1224-1228

[3] F. Metzger et al., “An Introduction to Online Video Game QoS and QoE Influencing Factors”, in IEEE Communications Surveys & Tutorials, vol. 24, no. 3, pp. 1894-1925, third quarter 2022,

[4] S. S. Sabet et al., ”Towards the Impact of Gamers Strategy and User Inputs on the Delay Sensitivity of Cloud Games”, 2020 QoMEX 2020, pp. 1-3,

[5] H. Sun, Research on Latency Problems and Solutions in Cloud Game, Journal of Physics: Conference Series. 1314. 012211, 2019.

[6] https://www.abya.tech/es, 2022.

[7] ITU-T Recommendation G.1032. Influence factors on gaming quality of experience, 2017.

[8] S. Zadtootaghaj et al., ”Modeling Gaming QoE: Towards the Impact of Frame Rate and Bit Rate on Cloud Gaming,” 2018 QoMEX, 2018, pp. 1-6,