Architecture analysis and specification of the RacingDrones platform: Online Video Racing Drones

César Benavente-Peces¹, David Tena-Ramos¹, and Ao Hu¹

Universidad Politécnica de Madrid, ETSIS Telecomunicación, Campus Sur, Calle de Nikola Tesla sn, 28031 Madrid, Spain,

`cesar.benavente@upm.es`,

`http://www.upm.es`

Abstract. This research work is aimed at the specification and development of a platform that allows online participation of multiple users and take control of a car (Racing-Drone). The management of the number of participating users will be analyzed according to the quality that the system can offer. Likewise, the different types of connection that can be used to guarantee the highest possible quality of service will be analyzed. This research has as its final objective the specification of the on-line multi-user platform to be developed to support the RacingDrones project. On the one hand, it will be necessary to study the architecture of the system in order to support the platform's engine and user access, taking into account aspects such as the capacity of the system. The software necessary to support the management of the game and the access of the users will also be analyzed. It will be necessary to carry out an analysis of the communications framework that must be developed in order to support multiple on-line players with the appropriate quality, managing the access permissions, the access requests and the maintenance of the connection. The most suitable protocols will be analyzed for the different functionalities of the system. After carrying out the analysis of all the aspects related to the platform (hardware, software and communications), the complete specification of the platform will be made. The devices from which users can access the platform will also be taken into account in the system specification. ...

Keywords: Drones, video streaming, video coding, remote driving, virtual reality

1 Introduction

Videogames, from the beginning, follow the footsteps of the film industry, with scripts increasingly worked, protagonists with their own personality, and a soundtrack performed by a renowned symphony orchestra. But in terms of the visual aspect, despite tending to photo-realism, they have always been somewhat behind, due to the difficulty in capturing a scene with the same fidelity as a real camera would.

One of the little explored possibilities to provide real sensations to the player is the use of real environments in a videogame, and instead of generating the images by computer, use a direct video recording that generates a competition vehicle. Although the result would be perfect in its photo-realism (it is real), we are faced with the technical problem of video coding in real time without affecting the game-play, which is a challenge and a problem not solved at present. In addition, in the case of mounting a video compressor inside a competition vehicle, the computer would be limited in size and weight and, consequently, in processing capacity, which makes the challenge a difficult problem to solve with the available technologies. The market for this type of technological solution would be very promising and is not limited to video games, as an important market that would have a tractor effect on this technology is the modeling competition market, where realism is not sought that can "simulate" a computer, but the same reality, for which high performance coding technologies are necessary.

Through a high-performance video coding technology solution, modeling competitions (and drones, both terrestrial and aerial) can be taken to the online world and take advantage of the potential of the synergy that both markets offer.

Within the modeling market, there are numerous circuits in cities around the world, and their exploitation is limited to specific meetings where competitions are held. On the contrary, in online games, every day there are ad-hoc competitions of players competing thanks to the online capacity of the games. Uniting both worlds could lead the modeling market much further, but for this a new technology is necessary.

In this context the main goal of this investigation is the analysis and definition of an appropriate architecture to provide real-time gaming facilities with a high QoS and a higher user experience.

The remaining part of this paper is structured as follows: Section 2 introduces the constraints and possible solution. The analysis of different solutions is carried out in sections 3, 4 and 5. The results obtained along this investigation are discussed in 6. Finally, some concluding remarks are provided in Section 7.

2 Analysis of possible solutions

In this section, the proposed solutions for the design and implementation of the platform are analyzed, taking into account the different aspects that may affect the quality of the online game service offered. One of the aspects that most impacts on the quality of the video-game service is latency. The aspects of the platform design that most commonly impact, to a greater or lesser extent, the latency of the platform are the following:

- The latency of the camera.
- The latency of the encoder.
- The technology of access to the network and the topology of it.
- The distance between the ends in which the communication is established, as well as the networks or intermediate nodes through which the data packets are routed.

- Computation capacity of the devices that are interconnected to process the data packets and generate the data packets from the video information at the transmitting end and receive the data packets at the receiving end and shape the video.
- The latency of the decoder.
- The latency of the screen.

There are some aspects which can not be influenced by an adequate design of the platform, such as the latency of the camera or the screen, which depend on the kind of device used and the manufacturing technology. Other aspects that influence the global latency will have higher or lower impact depending on how the architecture of the platform will be and what the specifications of the platform will determine to guarantee the best possible quality. There are other aspects such as jitter and lag produced in the transmission that are influenced by the network architecture (communications) that is used.

2.1 GAW vs. other solutions

GAW vs. other solutions. In order to evaluate GamingAnywhere [1], they have compared the performance of GamingAnywhere with OnLive [2] and Stream-MyGame (SMG) [3]. The experimental configuration consists of a server, a client and a router. The OnLive server resides in the OnLive data centers, while the GamingAnywhere and SMG servers are installed on our own PCs. More specifically, the OnLive client connects to the OnLive server over the Internet, while the GamingAnywhere and SMG clients connect to their servers through a LAN. To evaluate the performance of gaming systems in the cloud under various network conditions, we added a FreeBSD router between the client and the server, and executed dummynet to inject restrictions on delays, packet losses and network bandwidth.

Because the OnLive server is outside our LAN, the quality of the network path between our OnLive client and the server can affect our evaluations. However, according to our observations, the quality of the route was consistently good during all the experiments. The delay of the route network was around 130 ms with few fluctuations. In addition, packet loss rates were measured to less than 10^{-6} when OnLive transmissions were received at the recommended 5 Mbps. Therefore, the route between the OnLive server and our client can be considered as a communication channel with sufficient bandwidth, zero packet loss rate and a constant latency of 130 ms.

Since the performance of the game systems in the cloud can depend on the game, we consider games of three popular categories: action adventure, first person shooter and real time strategy. A representative game of each category is selected:

- LEGO Batman: The Videogame (Batman) [4] is an action adventure game created by Traveler's Tales in 2008. All the interactive objects in this game are made of Lego bricks. In this game, players control the characters to fight against enemies and solve riddles from a third person perspective.

- F.E.A.R. 2: Project Origin (FEAR) [5] is a first-person shooter game, developed by Monolith Productions in 2009. The combat scenes are designed to be as close as possible to those in real life. In this game, players have a great freedom to interact with environments, for example, they can flip over a desk to take cover.
- Warhammer 40,000: Dawn of War II (DOW) [6] is a real-time strategy game developed by Relic Entertainment in 2009. In campaign mode, players control squads to fight enemies and destroy buildings. In multiplayer, up to 8 players play matches on the same map to complete a mission, such as having specific positions.

Modern video encoders strive to achieve the highest quality video with the lowest bitrate by applying complex coding techniques. However, too complex coding techniques are not feasible for real-time videos given their long coding time. As such, empirically we studied the compensation between bit rate, video quality and frame complexity using x264. More specifically, we applied the coding parameters in real time, and exercised a wide spectrum of other coding parameters. Then they analyzed the quality of the resulting video and the coding time. According to the analyzes obtained in the study, the following x264 coding parameters are recommended:

- `-profile main -preset faster -tune zerolatency`
- `-bitrate $r -ref 1 -media -merange 16`
- `-intra-refresh -keyint 48 -sliced-threads`
- `-slices 4 -threads 4 -input-res 1280x720,`

where r is the coding rate. The GamingAnywhere server is configured to use the encoding parameters mentioned above, and the coding bit rate is set to 3 Mbps. For a fair comparison, all games are transmitted at a resolution of 720p. While we configure GamingAnywhere and OnLive to transmit at 50 fps, StreamMyGame only supports transmission at 25 fps. Experiments are designed to evaluate the three game systems from two critical aspects: response capacity and video quality. Experiments are also carried out to quantify the network loads incurred by different gaming systems in the cloud. The details of the experimental designs and the results are given below.

We define the response delay (RD) as the time difference between a user sending a command and the corresponding action in the game that appears on the screen. The RD is divided into three components:

- The processing delay (PD) is the time required for the server to receive and process the command of a player, and to encode and transmit the corresponding frame to that client.
- The reproduction delay (OD) is the time required for the client to receive, decode and represent a picture on the screen.
- The network delay (ND) is the time required for a round of data exchange between the server and the client. ND is also known as round trip time (RTT).

Therefore, we have $RD = PD + OD + ND$. Some remarks can be done regarding these parameters. First, the OD is small, $\leq 31ms$, for all cloud-gaming systems and considered games. This reveals that all the decoders are efficient, and the decoding time of the different games does not fluctuate too much. Second, GamingAnywhere achieves a much smaller PD, at most 34 ms, than OnLive and SMG, which are observed up to 191 and 365 ms, respectively. This demonstrates the effectiveness of the proposed GamingAnywhere: OnLive and SMG PDs are 3+ and 10+ times longer than those of GamingAnywhere. Load, among the three systems, only GamingAnywhere achieves RD less than 100 ms, and can meet the strict delay requirements of network games [7].

The server and client of GamingAnywhere are well paired, in the sense that all the steps in the production line are quite efficient. Even for the video coding that consumes more time (on the server) and video playback (on the client), each frame ends at an average of 16 and 7 ms maximum. Such low delay contributes to the superior performance RD of GamingAnywhere, compared to the other known cloud gaming systems.

2.2 Video quality

Video streaming quality directly affects the gaming experience, and network conditions are the keys to high-quality streaming. From this point of view, we use dummynet to control three network condition metrics: network delay (ND), packet loss rate and network bandwidth. We vary ND between 0-600 ms, packet loss rate between 0-10% and bandwidth 1-6 Mbps in the experiments. Also included are experiments with unlimited bandwidth. For OnLive, the ND on the Internet is already 130 ms and, therefore, you can not report the results from scratch ND. Two video quality metrics, PSNR and structural similarity (SSIM), are adopted. The average values of PSNR and SSIM of component Y are reported.

The proposed GamingAnywhere solution incurs much lower uplink traffic loads, compared to OnLive and SMG. The only exception is that, with Batman, SMG incurs a lower rate of uplink packets. However, SMG also produces a larger uplink payload size, which leads to a higher uplink bit rate than GamingAnywhere. The downlink bit rates of OnLive are between 3-5 Mbps, while those of SMG are between 10 and 13 Mbps. This finding indicates that the compression algorithm used by OnLive achieves a compression rate up to 4.33 times higher than that of SMG.

The values of PSNR and SSIM, respectively. We make four observations about these two figures. First, ND does not affect the quality of the video too much. Second, GamingAnywhere achieves much higher video quality than OnLive and SMG: up to 3 dB and 0.03, and 19 dB and 0.15 spaces, respectively. Third, GamingAnywhere suffers quality declines when the packet loss rate is not trivial. This can be attributed to the lack of fault tolerance mechanism in GamingAnywhere. However, high packet loss rates are less common in modern networks. Finally, the video quality of GamingAnywhere drops suddenly when the bandwidth is less than the coding bit rate of 3 Mbps. A possible future

work to address this is to add a speed adaptation heuristic to dynamically adjust the rate of coding bits, in order to use all the available bandwidth without overloading the networks.

3 Proposal #1

In this first proposal shown in Figure 1 the generic architecture of the platform includes the use of GAW in the car, running in a Raspberri PI, and with GAW in the end of the client (player). In this possible solution the GAW would be embedded / connected to the FPGA in order to optimize the overall quality of the services. The LHE encoder would be embedded in the FPGA to optimize performance and minimize coding time. In this case, an ad-hoc software would have to be developed in the server as a videogame that forwards the commands produced by the player to the car.

3.1 Advantages

The main advantage of this architecture design of the platform is that its implementation is very simple.

3.2 Disadvantages

The main drawback in this scheme is that the frame is retained in GAW until it is complete, so time is lost in this task increasing the overall latency of the game.

4 Proposal #2

This solution shown in Figure 2 would include the development of the following subsystems:

- development of an ad-hoc software at the end of Racingdrone to capture the video, compress it efficiently (fast and lightweight) and send it to the server.
- development of an ad-hoc software for the sending of the racingdrone control commands.
- development of ad-hoc software at the end of the client client.

4.1 Advantages

It is not required in this case to retain the frame in the car until it is complete, so that the overall latency is reduced.

Fig. 1. Proposal #1 of system architecture

4.2 Disadvantages

In this solution proposal you have to program everything, since there would be no existing software that could carry out some of the necessary functionalities, software of the RPI, create a kind of proxy on the server side of video games, and also on the side client.

5 Proposal #3

In this possible solution shown in Figure 3 the GAW is implemented in a server, but it does not carry out a transcoding, but it is used only for the handling of the control commands of the racingdrones and to forward the video flow to the authorized players.

In this possible solution it would be necessary to develop an ad-hoc software in RPI to capture the video. In addition, it would be necessary to develop ad-hoc software on the server as a videogame that forwards the commands to the car from the player interface.

5.1 Advantages

As advantages of this solution proposal are, first of all, that the frame is not retained on the side of the car, improving with this option the overall latency

Fig. 2. Proposal #2 of system architecture

time of the system (taking into account all the factors that influence it) and, On the other hand, at the end of the client it is possible to make use of solutions that are already implemented, such as the case of the GAW Client, which although it is necessary to develop an integration work with the other components of the platform, allows to advance in the fastest development.

5.2 Disadvantages

The main drawback that appears in this solution is that the frame is retained in the server, although we have the intuition (and that its veracity will be analyzed) that it is not necessary to retain the frame in the server since it can leave the car in RTP, which it is the optimal solution that will be developed.

6 Results

Based on the analysis that has been carried out in sections 3, 4 and 5, the solution adopted, which guarantees the best quality of service and playability, is *Proposal #3*. In the decision of the choice of said solution have fundamentally influenced the following aspects: Latency: this solution is the one that provides the lowest overall latency time, as previously mentioned, because the frame is not retained at the end of the racingdrone, where the computing capacity is more compromised. In any case, it must be produced on the server. Solutions

Fig. 3. Proposal #2 of system architecture

on the client side: there are solutions for the end of the client that are already developed and efficient, although it is necessary to carry out a rather intense work to be able to carry out the integration with the rest of the elements of the platform.

7 Conclusion.

This research has the objective of analyzing the different options of platform implementation, highlighting the advantages and disadvantages of each of them in order to determine and finally propose the design specifications of the platform. We have analyzed three possible implementation schemes, each of them with particular implementation requirements that influence the total quality of service (QoS) offered to the user (player). In the decision of the solution adopted, various factors have influenced it, ultimately, impact on the quality of service offered to the user. Among these aspects is the latency time, which is influenced by aspects not only related to the quality of the communication link, but also others related to coding/decoding and in particular the influence of where this process is carried out, since It is very influential on the total latency. The use of the LHE will greatly improve the coding/decoding latency compared to other video encoders.

The use of an FPGA in which the LHE encoder is embedded will improve the coding latency and, therefore, the overall platform.

The impact of the latency on the manageability of the racingdrone has also been taken into account, so that the reaction times allow a real-time control of the vehicle that provides the user with a satisfactory experience in the game.

After carrying out the analysis of the different options taking into account the restrictions they impose, it has been found that the most optimal solution is *Proposal #3*, whose diagram was shown in Figure 3 and based on this, the specifications of the platform were defined.

Depending on the system architecture and the aspects that affect the QoS, the limit values of the parameters that must be preserved have been determined so that the player has a good experience in the development of the game.

Acknowledgments

The authors wish to thank the Spanish Science and Innovation Ministry which funds this research through INNPACTO innovation program RTC-2016-4744-7.

References

1. <http://gaminganywhere.org/perf.html>
2. <http://gaminganywhere.org/perf.html\#onlive>
3. <http://gaminganywhere.org/perf.html\#smg>
4. <http://gaminganywhere.org/perf.html\#batman>
5. <http://gaminganywhere.org/perf.html\#fear>
6. <http://gaminganywhere.org/perf.html\#dow>
7. <http://www.gaminganywhere.org/perf.html\#CC06>