

Thesis no: MSEE-2016-53



On the Quality of Service of mobile cloud gaming using GamingAnywhere

Veera Venkata Santosh Surya Ganesh Grandhi

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Masters in Electrical Engineering with emphasis on Telecommunication Systems. The thesis is equivalent to 20 weeks of full time studies.

This master thesis is typeset using L^AT_EX.

Contact Information:

Author:

Veera Venkata Santosh Surya Ganesh Grandhi

E-mail: vegr15@student.bth.se,
grandhiganesh55@gmail.com

University advisor:

Dr. Yong Yao

Department of Communication Systems

University examiner:

Dr. Siamak Khatibi

Department of Communication Systems

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

In the recent years, the mobile gaming has been tremendously increased because of its enormous entertainment features. Mobile cloud gaming is a promising technology that overcome the implicit restrictions such as computational capacity and limited battery life. GamingAnywhere is an open source cloud gaming system which is used in this thesis to measure the Quality of service of mobile cloud gaming. The aim of thesis is to measure the QoS used in GamingAnywhere for mobile cloud gaming. Games are streamed from the server to the mobile client. In our study QoS is measured using Differentiated Service (DiffServ) architecture for the traffic shaping. The research method is carried out using an experimental testbed. Dummynet is used for traffic shaping. Performance is measured in terms of bitrate, packet loss, jitter and frame rate. Different resolutions of the game are considered in our empirical research and our results show that the frame rate and bitrate have increased with the impact of network delay.

Keywords: Mobile cloud gaming, Performance Evaluation, Quality of Service(QoS)

Acknowledgments

This thesis work could not have been possible without our supervisor **Dr. Yong Yao**, who encouraged and challenged me throughout the journey. He always motivated me to give the best effort possible.

I would like to express my heartfelt gratitude to him. It is a matter of great privilege to carry out my thesis under his supervision. He has always inspired and guided me in the right direction throughout the way.

And also, I would like to thank examiners and professors of DIKO, **Dr.Siamak Khatibi**, **Dr.Kurt Tutschku** and **Dr.Markus Fiedler** for their suggestions in the thesis and ATS course respectively.

Most importantly I would like thank my family for their support, and especially Suren for sharing his contribution in the completion of thesis. Special thanks to Umesh brother for sharing his knowledge and guidance. Also I would like to thank Phani and my close friends for encouraging and supporting me.

Finally, I thank almighty Gods. Without their blessings, this journey would not have been possible for me.

-Ganesh Grandhi
October, 2016

Dedicated to my beloved grandfather...

Contents

Abstract	i
1 Introduction	1
1.1 Motivation	2
1.2 Aims and Objectives	3
1.3 Research Questions	3
1.4 Research Methodology	3
1.5 Thesis Outline	4
1.6 Split of work	5
2 Background	7
2.1 Video Gaming	7
2.2 Overview of Cloud Gaming	7
2.3 Mobile Cloud Gaming	9
2.4 Existing Cloud Gaming	9
2.5 Architecture of GamingAnywhere	10
2.5.1 GA server	10
2.5.2 GA client	10
2.5.3 Interaction between the GamingAnywhere server and client	11
2.6 Quality of Service	11
2.6.1 ITU/ISO	12
3 Related Work	14
4 Methodology	16
4.1 Research Technique	16
4.2 Evaluation Technique	16
4.3 Experimental Testbed	17
4.3.1 Testbed Design	17
4.3.2 GamingAnywhere	18
4.3.3 AssaultCube	20
4.3.4 Test Enclosure	20
4.3.5 Approach to the QoS	20

5	Experimentation and Results	23
5.1	Experimental Procedure	23
5.2	Trace Collection	24
5.3	Experimental Scenarios	24
5.3.1	Scenario 1	24
5.3.2	Scenario 2	25
5.3.3	Scenario 3	25
5.3.4	Scenario 4	26
6	Analysis and Discussion	27
6.1	Bitrate performance	27
6.2	Packet Loss performance	28
6.3	Jitter performance	29
6.4	Frame Rate performance	30
7	Conclusions and Future Work	31
7.1	Conclusion	31
7.2	Future Work	32
	References	34
	Appendix	37

List of Figures

2.1	Cloud Gaming Overview	8
2.2	GamingAnywhere system Architecture	11
2.3	Four view points of quality of service[21]	12
4.1	Experimental test-bed	17
4.2	GA Client mobile application	19
4.3	Assault cube game running on mobile device	19
5.1	A Demonstration of experimental setup	23
6.1	Performance of bitrate under different network conditions	27
6.2	Performance of packet loss under different network conditions	28
6.3	Performance of jitter under different network conditions	29
6.4	Performance of frame rate under different network conditions	30

List of Tables

1.1	Split of Work	5
4.1	Technical specifications	18
5.1	Average bandwidth for different resolutions	24
5.2	QoS metrics for ideal case	24
5.3	QoS metrics for 0.1% packet loss	25
5.4	QoS metrics for 0.3% packet loss	25
5.5	QoS metrics for 33ms delay	25
5.6	QoS metrics for 63ms delay	26
5.7	QoS metrics for bandwidth of 2000 Kbps	26
5.8	QoS metrics for bandwidth of 2500 Kbps	26

List of Abbreviations

AP	Access Point
FPS	Frames Per Second
GaaS	Gaming as a Service
GA	Gaming Anywhere
GUI	Graphical User Interface
ITU-T	International Telecommunications Unit, Telecommunications Sector
ISO	International Organization for Standardization
MCG	Mobile Cloud Gaming
QoS	Quality of Service
PC	Personal Computer
QoE	Quality of Experience
QoS	Quality of Service
RF	Radio Frequency
RTP	Real Time Transport Protocol
RTCP	Real-time Transport Control Protocol
RTSP	Real Time Streaming Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WLAN	Wireless Local Area Network

Chapter 1

Introduction

From the past few years, video gaming is one of the quickest developing business in interactive media amusement industry. Filled by the extensive scale reception of smart mobile devices, mobile game such as Pokemon Go have shocked numerous by hitting colossal number of players in a short time frame. In spite of huge development of ability in mobile devices, gaps stay between the support of mobile devices and necessities of incredible video games. High end video game scenes often need intensive graphics and high computational energy to render sensible and intuitive gaming scenes[13]. When compared to PC and console games, mobile games are less visually attractive due to its restricted requirements like graphics, memory, network bandwidth and battery limit. These attributes may push genuine gamers away from mobile games[20].

With the revolutionary of core and expansive broadband networks, many services were carried away by the remote data centers from the end devices. This is broadly referred to as cloud computing. As the number of remote services and applications for the users are increasing every day, the requirement of network Quality of Service is increasing to meet the norms of the user[22]. Cloud gaming, a kind of cloud service also referred as Gaming as a Service (GaaS) is a new technology which is born by the fruitful ideas of both cloud computing and online gaming.[28][29].

“Cloud gaming renders a gaming application remotely in the cloud and streams the video scenes back to the user over the internet.[28]” Mobile cloud gaming has been prevalent in recent times due to the advancement in cloud computing and mobile communications. Mobile Cloud Gaming (MCG) is a promising innovation which has been enabled to address the intrinsic confinements of mobile devices[10].

Mobile cloud gaming system has many benefits: (i) Better visual quality attract serious gamers, (ii) Lower porting cost/effort can play more games, (iii) Low battery consumption gives longer play time. Since the design space of cloud gaming system is large, several challenges poses in the area of Quality-of-Service (QoS) management. QoS is basic fundamental for cloud gaming users,

who expect providers to deliver the quality characteristics, and for cloud gaming providers, who need to find trade-offs between QoS levels and operational expenditure[20][11].

“In cloud gaming, the games are stored in the remote datacenter’s platform and are streamed to the end users devices. Onlive(bought by Sony) , Stream-MyGame, Gaikai are the closed and proprietary gaming companies which offer cloud gaming services.

GamingAnywhere[4] is the first open-source cloud gaming system. It has a cross-platform functionality. It is supportable to Windows, Linux, OSX, and Android. Thus, GamingAnywhere can be used by researchers, end users, developers, service providers for setting up cloud gaming testbed. This GamingAnywhere can port the cloud gaming client to mobile devices[19][20].

The performance of cloud gaming may be game-dependent. In general context, the three popular categories of games present are action adventure, real-time strategy and first person shooter. In our research, first person shooter game is considered. The performance is calculated for individual game sessions.

The performance of GamingAnywhere can be evaluated from different views. By considering the perspective of service providers, the efficiency of resource allocation scheme is important and for the gaming experience, the QoS metrics are important for perspective of end users. Time-Scale is one of the way to classify the performance metrics. Larger time-scale metrics are used for multiple game sessions. Smaller time-scale metrics are used for individual game sessions[11].

This thesis is focused on smaller time-scale metrics from the end users perspective. The streaming communication network between the server and the mobile client is also one of the problem that affects the gaming experience of the user. The QoS is measured in terms of performance metrics like network band-width, delay, jitter and packet loss. In mobile cloud gaming, thin clients (referred to as mobile devices) are connected to cloud servers via wireless networks such as Wi-Fi, 3G, LTE, etc., These connections may incur network delay. This thesis is mostly focused on to implement this mechanism on a test bed and study its performance.

1.1 Motivation

From the past five years, mobile gaming has been advanced a lot. The computational load of a mobile device when playing a game, should be offloaded to server so as to save energy and time on terminal side. As a result, the cloud gaming came into existence. As many cloud gaming systems are proprietary and closed, GamingAnywhere is the only open-source cloud gaming system available

for gamer, developer or service provider. The reason that mobile game player demands a certain level of quality of service(QoS) is to get a good gaming experience. This thesis is motivated by the question: How good is the QoS of mobile cloud gaming using GamingAnywhere?

1.2 Aims and Objectives

The aim of this thesis is to know how is the Quality of Service(QoS) of mobile cloud gaming system using GamingAnywhere. The performance metrics will be analyzed and compared for different network scenarios. The results will be documented in our thesis report. The following are the objectives that are to be implemented to achieve the goal.

- Understand the GamingAnywhere system and its implementation with the mobile.
- Identify the performance metrics to be evaluated for quality of service for GamingAnywhere.
- Perform with a game so as to understand the GA mobile client performance.
- Set up the experimentation environments for criterion of evaluation.
- Compare the performance metrics of GamingAnywhere in different network scenarios.

1.3 Research Questions

1. How much network bandwidth is utilized by a game session?
2. How can we achieve trade-off for QoS for different screen resolutions with the impact of network delay?
3. What is the impact on QoS metrics when subjected to variable packet loss in the network?
4. What is the impact on QoS metrics when the network bandwidth is limited?

1.4 Research Methodology

1. In the early phase of our research, thorough study of GamingAnywhere is done and then a detailed study of Quality of Service, the metrics to be evaluated.

2. Study and analyze what are the performance metrics need to be evaluated for the quality of service.
3. In the next stage, installation, configuration and implementation of GamingAnywhere testbed is studied and done where GA server and GA client are installed in our PC and mobile respectively.
4. Different games are studied and played in the server and are streamed to the client. Different scenarios are made to obtain better QoS under different available conditions.
5. The performance metrics must be evaluated by capturing the network traffic under available network connection types.
6. The tools used for evaluating the performance are studied and analyzed how to configure them.
7. Different experiments are conducted under different scenarios in different environments.
8. Results are noted and observed and later analyzed. Based on final results, conclusions and recommendations are provided.

1.5 Thesis Outline

Chapter 1: Introduction briefly explains about the cloud gaming and mobile cloud gaming technologies and how they are being prominent these days. GamingAnywhere an open cloud gaming system depiction is also mentioned.

Chapter 2: Background describes about the video gaming and its antiquity. A brief overview of cloud gaming and existing cloud gaming systems were explained. After that, the architecture of GamingAnywhere is described and explained with the help of the figures.

Chapter 3: Related Work comprises of the contributions by other authors who had done their work on topics such as cloud gaming, GamingAnywhere, Quality of Service on cloud gaming. It depicts about their implementations and how it motivated research.

Chapter 4: Methodology gives a comprehensive demonstration of the approach. It explains about the experimental test bed, design of GamingAnywhere, Quality of Service model and its measurement analysis is explained.

Chapter 5: Experimentation and Results outlines the experimental procedure and results are tabulated for each experimental scenarios .

Chapter 6: Analysis and Discussion section displays the collected data in graphical format. It depicts the different performance metrics conducted in the research and analysis of the results. In-depth reasoning is given about the obtained results.

Chapter 7: Conclusion and Future work section concludes the thesis. It outlines the results and also give the suggestions for the future work that can be done.

1.6 Split of work

This section illustrates the distribution of work among the thesis partners.

SECTION	TOPIC	CONTRIBUTOR
Chapter 1	1.1 Motivation	Veera Venkata SSG Grandhi Suren Musinada
	1.2 Aims and Objectives	Veera Venkata SSG Grandhi
	1.3 Research Questions	Veera Venkata SSG Grandhi
	1.4 Research Methodology	Veera Venkata SSG Grandhi Suren Musinada
	1.5 Thesis Outline	Veera Venkata SSG Grandhi
Chapter 2	2.1 Video Gaming	Veera Venkata SSG Grandhi Suren Musinada
	2.2 Overview of Cloud Gaming	Veera Venkata SSG Grandhi Suren Musinada
	2.3 Mobile Cloud Gaming	Veera Venkata SSG Grandhi Suren Musinada
	2.4 Existing Cloud Gaming	Veera Venkata SSG Grandhi Suren Musinada
	2.5 Architecture of GamingAnywhere	Veera Venkata SSG Grandhi Suren Musinada
	2.6 Quality of Service	Veera Venkata SSG Grandhi
Chapter 3	Related work	Veera Venkata SSG Grandhi
Chapter 4	4.1 Research Technique	Veera Venkata SSG Grandhi
	4.2 Evaluation Technique	Veera Venkata SSG Grandhi
	4.3.1 Testbed Design	Veera Venkata SSG Grandhi
	4.3.2 GamingAnywhere	Veera Venkata SSG Grandhi Suren Musinada

Continued on next page

Table 1.1 Split of Work – *Continued from previous page*

SECTION	TOPIC	CONTRIBUTOR
	4.3.3 AssaultCube	Veera Venkata SSG Grandhi
	4.3.4 Test Enclosure	Veera Venkata SSG Grandhi Suren Musinada
	4.3.5 Approach to QoS	Veera Venkata SSG Grandhi
Chapter 5	Experimentation and Results	Veera Venkata SSG Grandhi
Chapter 6	Analysis and Discussion	Veera Venkata SSG Grandhi
Chapter 7	Conclusions and Future Work	Veera Venkata SSG Grandhi

2.1 Video Gaming

Single – player computer games began spreading with adoption of PCs in the 1980s. With the expansion of the internet and networked society, computer games progressively incorporated multiplayer features. As mobile gadgets are being increased, mobile games came to the real world. Recently, the growth of cloud computing has quickened the ascent of GaaS. The GTA IV computer game collected \$310 million within a span of 24 hours around the world in April 2008. The video game industry is anticipated to reach \$82 billion by 2017. Indeed, in depressed economy also, video gaming industry growth has been increased by 16%. The famous game distributor Electronic Arts earned 40% of income through online. This shows that video games are not only for entertainment but also intensely rely in online infrastructure. The cloud gaming technologies majorly have social gaming, multiplayer online games and many cloud services. Cloud services (for example, XBOX live) are just intended to increase the functionalities of user gadgets with social networking and finite storage with server access[29].

2.2 Overview of Cloud Gaming

Cloud computing has extremely changed existing operations and business models of IT industry due to its unparalleled versatility and reduced expenses of capital and hardware support. Existing applications, from file sharing to multimedia streaming, have encountered an awesome benefit from cloud computing platforms, in terms of system/framework efficiency and its ease of use. The advancement of cloud computing extends to allow offload complex tasks like intensive graphics 3D rendering to the cloud. This has turned the possibility of cloud gaming into a reality and significantly facilitates the development[31].

Since 2009[26] to present, cloud gaming is referred as: *“Cloud gaming is an online gaming where games and their logics are stored in the servers and the video game scenes are streamed back to the thin client over the Internet. The thin*

client is responsible for displaying the video from the cloud rendering server as well as collecting the player's commands and sending the interactions back to the cloud[28]."

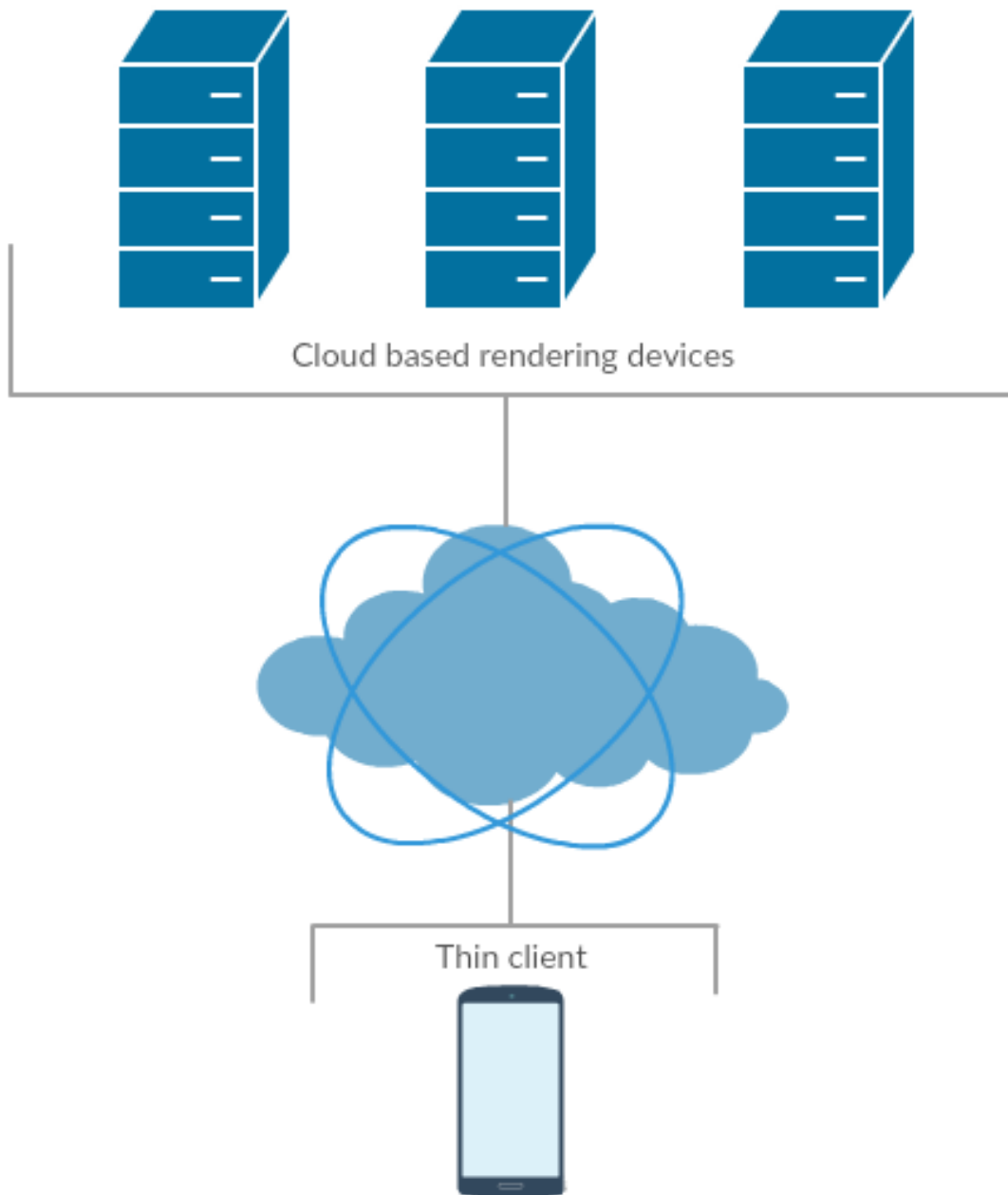


Figure 2.1: Cloud Gaming Overview

2.3 Mobile Cloud Gaming

Mobile cloud gaming is a new and promising technology for the cloud-based delivery of video games to mobile devices in a platform-independent manner. These mobile devices can be smartphones and as well as tablets.

Mobile cloud gaming is defined as communal gaming utilizing smartphones that access the cloud as an outer source for working of gaming scenarios and their interactions, and also to empower the extreme features like cross-platform operations and battery saving[10].

2.4 Existing Cloud Gaming

Cloud gaming has obtained a lot of significance and so many companies showed interest to provide several cloud gaming services. In 2012, Sony computer entertainment attained Gaikai[3] which was founded in 2008. It is the earlier lead company to provide cloud gaming services. Sony computer entertainment has built up a high quality and rapid cloud streaming platform that helps in rapid delivery of games and more associative services to the customers over the web. The main disadvantage in using Gaikai is that it does not tablets and digital TVs on the client side.

There is an another cloud gaming service called Onlive[5] game from Onlive corporation which invented high graphic applications that are available for all devices such as laptops, PCs and tablets. For handling these type of high resolution applications, H.264 encoder is configured on the server side. However, it uses Virtual machines on server side which in turn gives two streams. One stream is for gaming in the real world which is said to be the live stream and the other stream is for recording the game sequences likely said to be media streaming.

StreamMyGame[6] is a closed cloud gaming system which is quite similar to Onlive is only software game streaming solution that plays the Microsoft games remotely on Linux and Windows platforms with great number of users. However, the system layout cannot be examined as this is a closed cloud gaming system[31].

GamingAnywhere is the only cloud gaming system which is open to all. Compared to the remaining closed cloud gaming systems, GamingAnywhere utilizes the video streaming methodology for lower loads on the client side[19]. GamingAnywhere includes extensibility, portability, configurability and openness. It embraces a modularized plan. Video and audio capturing which are platform dependent segments and platform independent segments such as network protocols and codecs can be effortlessly changed and supplanted. Game users can utilize

their gadgets with heterogeneous designs to access GamingAnywhere and so it can be ported to any platform. Modern platforms can be effectively upheld by supplanting the platform dependent segments in GamingAnywhere. A vast number of implicit video and audio codecs are supported by GamingAnywhere. It has given all possible configurations to the game users so that they can try the combinations of the parameters in different situations.

2.5 Architecture of GamingAnywhere

GamingAnywhere comprises of two components. They are GamingAnywhere server and GamingAnywhere client.

2.5.1 GA server

The selected game runs on the game server along with an agent. The agent is a thread injected into the selected game. The first requirement of the agent is to capture the video and audio frames that are produced by the chosen game and encode the frames using chosen codecs[23]. These frames are delivered to the client via the data flow. The second requirement is to communicate with the game dynamically. As the user sends actions from the client, the server must behave, as though it was the user and must re-play the received input events[23].

The server operation involves two main mechanisms to capture game screens. The first mechanism performs a screen capture of the entire desktop periodically, then extracts the particular region associated with the game screen. After this, the second mechanism is that the video frame data is obtained from the graphic rendering buffer[23]. This method is executed by using the hook function of DirectX APIs. By using the Window Audio Session API, the audio frames were captured for the game sound. After getting the audio and video frames on server side, the encoding is done by utilizing the libavcodec library. Then the encoded information is streamed to client from the server.

2.5.2 GA client

The GamingAnywhere client decodes the information which is streamed from the server and render the video and audio frames. For decoding, the GA client also utilizes libavcodec library. Zero-buffering mechanism is used by the decoder. This mechanism is used to reduce the effect of delay while playing the game[23].

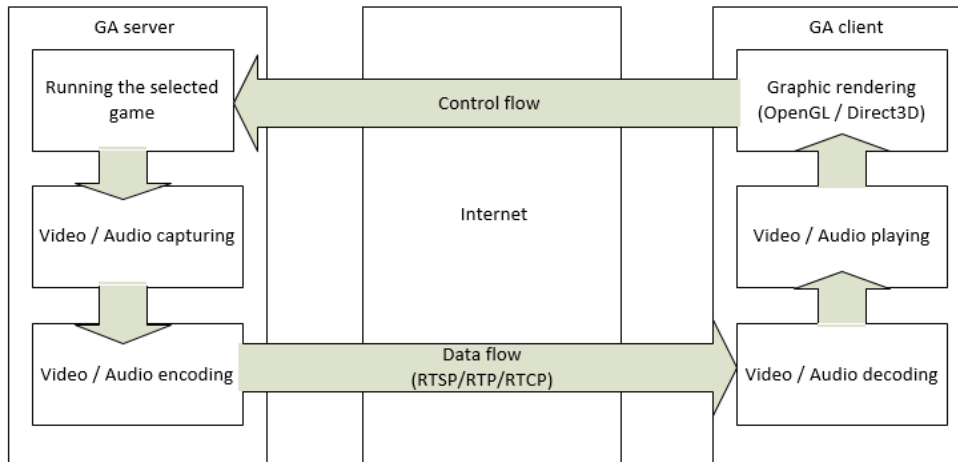


Figure 2.2: GamingAnywhere system Architecture

SDL library is designed to provide the real time rendering of the decoded video frames through Direct3D and OpenGL. The SDL library precisely read and render the encoded audio frames. After that, SDL library also take the user inputs such as the tapping of the touchscreen on mobile and the pressing of keys on keyboard and moving the mouse in case of Windows and linux. After capturing the user inputs, they are sent to server side from the client side through control flow.

2.5.3 Interaction between the GamingAnywhere server and client

In GamingAnywhere, the control flow and data flow establish the interaction between the GamingAnywhere server and client. For handling the control flow and data flow, the libraries such as libavformat and live555 are used on both sides[23]. The audio and video frames were streamed to the client from the server by using two ways. One is by transmitting RTCP/RTP packets through the RTSP over TCP and the other way is by transmitting the RTP packets over UDP. The GamingAnywhere client handles protocols such as RTSP/RTP corresponding to the method used on the server side.

2.6 Quality of Service

In general context, quality of service(QoS) is defined as the total performance of the computer network i.e the performance observed by the end users in the network. In telecommunication systems, Quality of Service(QoS) is defined in many

ways. The following is the standard relation for the QoS defined by International Telecommunication Union (ITU)[16].

2.6.1 ITU/ISO

The ITU has defined a set of suggestions in the area of QoS. Most of these suggestions has also written by the ISO. They cover wide distinctive ranges on the field of general QoS systems, QoS administration and measurement, QoS observed by the user and QoS identified with multimedia applications[16].

ITU-T standard E.800[21] defines QoS as *“the collective effect of service performance which determine the degree of satisfaction of a user of the service.”*

It also explained the concept QoS, its relation with the network performance and provided a set of performance measures particularly for the telecommunication networks. To understand QoS in a detailed manner[21], it can be partitioned in four viewpoints shown in the Figure 2.3.

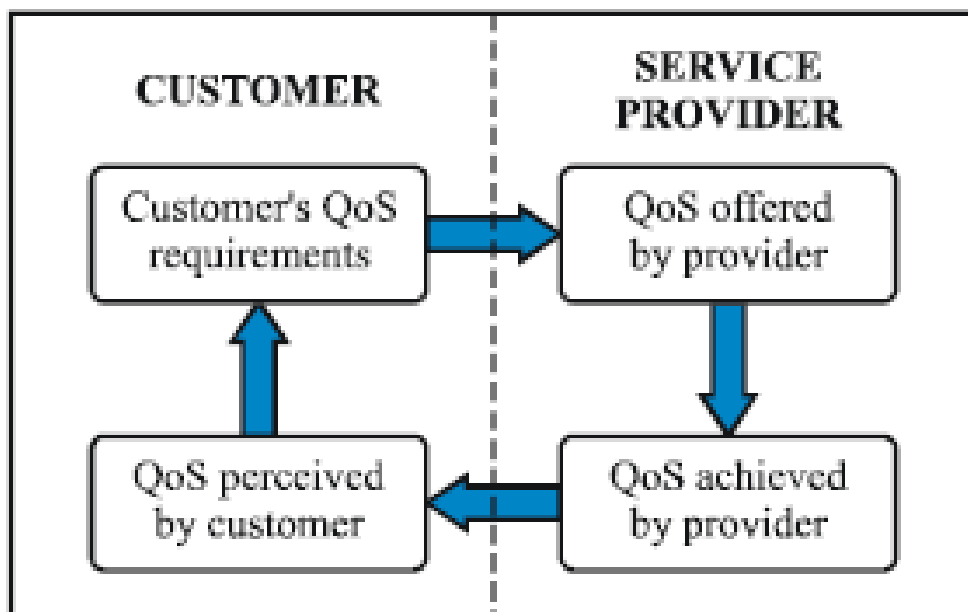


Figure 2.3: Four view points of quality of service[21]

QoS experienced is affected by the conveyed QoS and the psychological factors impacting the perception of the user. Understanding of the QoS experienced is essential to enhance income and resources of the service provider.

User’s opinion is the fundamental aspect while estimating a service. Thus, it’s the level of satisfaction with the provider that is the main entity it specifically

interacts with. This degree of satisfaction comes from the view of different service aspects. These service aspects include support, operability, serviceability and security. These aspects are affected by the network performance characteristics.

To maintain end-to-end QoS, service support and operability aspect is required in which the service is supported(connection oriented or connection less). Serviceability aspect interfaces with the network trafficability performance.

The QoS consists of both network and non-network related performance. For the quantitative measurement of QoS, network performance characteristics include bit rate, packet loss, latency, jitter, etc. Non-network performance characteristics include provision/repair time, complaints resolution time etc.

In [27] Authors have discussed necessities and challenges for QoE management in the case of cloud gaming. To assess the cloud gaming QoE, compared with the previous studies, the authors looked into how the additional network delay in cloud gaming situation is perceived by the gamers w.r.t delay identification, acknowledgment, type of game and also rating consistency. They presented a need of building a game portrayal plot that covers important related parameters. This work ponders the QoS requirements based on exact QoS/QoE mappings. Key factors that influence QoE/QoS mappings are also discussed. Authors have used Steam platform and Xbox 360 game controller with the client. To control network traffic, i.e to add additional network delay, Netem is used. Electronic questionnaire for the gamers is used to know their feedback when delay is ranged from 20 to 500. Their results reveal that the platform game has less impact compared to racing game for 500ms of delay, i.e game type affects the overall perceived quality. Up to 50%(total 35) of users have detected a delay when additional 200ms delay is set.

Authors of [18] elucidated that internet only provides better effort services and game users are hard to please. Despite researchers are improving cloud gaming systems but the existing cloud gaming systems were unable to reach the researchers ideas as they are closed and proprietary cloud gaming systems. So, Authors of [18] have presented first open source cloud gaming system GamingAnywhere, an extensible, configurable and portable cloud gaming system. The GamingAnywhere can be utilized by the engineers and researchers to try their new ideas in real world. It can also be used by the service distributors for developing services in cloud gaming and the game users to organize their own cloud gaming system. Authors of [18] had demonstrated GamingAnywhere system with one GamingAnywhere server and four GamingAnywhere clients and then they have compared the response delays of GamingAnywhere with the closed and proprietary cloud gaming systems StreamMyGame and OnLive. The authors also concluded that GamingAnywhere cloud gaming system can help the multimedia research community to establish and develop more cloud gaming systems in future.

Authors of [20] used GamingAnywhere and conducted experiments in cloud

gaming to quantify satisfaction of users. They have used both PC (Windows OS) and mobile (Android OS) clients for their experiments. User studies are conducted i.e gamers are called and played different games on both clients for one minute and evaluated their gaming experience. The evaluation is based on the aspects like smoothness, graphics and control given on a scale of five-level MOS. Each game session is varied with four different parameters like bitrate, frame rate, resolution and delay. Authors used games like Limbo, Mario Kart 64, Super Mario 64 and Super Smash Bros. Their experimental results show that users are more satisfied with the graphics on mobile devices compared to desktops and in case of smoothness PC is slightly higher, also more satisfied with the control quality on desktops. They also shared their experience of GA client to Android OS. They also analysed and explained which parameters affect the above aspects.

Authors of [11] come up with a set of techniques to evaluate the QoS of cloud gaming systems. They have analyzed Onlive and StreamMyGame and compared their QoS in terms of network traffic characteristics, frame rate, delay elements etc. Their research was based on small time-scale performance metrics from the users perspective. In our thesis, we adopted their evaluation technique for the performance of mobile cloud gaming. Their evaluation is performed on desktop computers and are concentrated on objective quality metrics. Free BSD 7.0 router is used for the traffic injecting between server and the client. Their results reveal that the OnLive performs better than StreamMyGame in terms of framerates, graphics quality, server processing delay, and also with less network bandwidth.

Chen et al.[12] defined latency, also known as a response delay is difficult to find out in cloud gaming. Response delay is defined as the combination of network delay, processing delay and playout delay. Different techniques have been followed by them. Authors considered Onlive and StreamMyGame to measure the latency in cloud gaming systems.

Authors of [24] described that mobile cloud gaming is the latest approach for the cloud based delivery of games to the mobile phones in a platform independent style. Authors have considered a service model considering energy consumption, latency and cost. Measuring the latency consists of many categories such as the game pipeline latency, the encoding and decoding time span and latency of network. In the experiment, network latency is measured by considering the common WiFi and cellular networks. From the analysis of latency, authors depicted that latencies somewhat increase when there is a slight increase in geographical distance which help the cloud gaming distributors to build datacenters close to the customers. Authors of [24] concluded that mobile cloud gaming based on UMTS will have some advantages on a local game execution on latest mobile devices having powerful hardware.

This chapter discusses the research methodology of the thesis. The techniques implemented to measure the QoS of mobile cloud gaming reaches the goals of the thesis. To achieve the necessary goals, the research technique has to be in par with the desired outcomes. This thesis requires the specific research to achieve QoS for mobile cloud gaming in different scenarios.

4.1 Research Technique

The different approaches available in modern research are Qualitative, Quantitative and both Mixed method approaches suggested by John W. Creswell[15]. Quantitative research method was followed in this thesis. This research technique mainly comprises of objective measurements and statistical analysis of data. The data is gathered from various sources with a specific end goal to sum up a particular phenomenon or concept. Statistical analysis is done with this data.

The results and analysis are exclusively based on numerical findings from experimental tests. The metrics calculated for QoS in different scenarios are the basis for our conclusions.

So, the quantitative method of research was utilized to achieve the QoS of mobile cloud gaming using GamingAnywhere.

4.2 Evaluation Technique

For the evaluation or studying of a system [25], experiments are performed with the model of the system. Physical and Mathematical are the two models available for the model of a system. Physical model is the realistic replica of the actual system. The physical system carries all properties and characteristics of actual system.

This research involves the experimentation with the physical model of the GamingAnywhere performance for mobile cloud gaming. The physical model is considered over an mathematical simulation model, because this thesis goal is to

implement GA on a physical system and to measure the QoS during experimentation

4.3 Experimental Testbed

This section discusses the software and hardware chosen for the experiment setup, to observe the performance metrics and experiments conducted in different scenarios. Mobile cloud gaming require both a server and client(as mobile) to function.

4.3.1 Testbed Design

The experiment environment consists of two hosts: GamingAnywhere server, GamingAnywhere mobile client, and RF shielded test enclosure along with WLAN AP. The WLAN AP is enclosed in a RF shielded test enclosure often called it as black box. A full duplex WLAN test network has been built up utilizing WLAN access point. The test bed set up is shown in the Figure 4.1. The GA mobile client and GA server were connected to the WLAN AP via 802.11 wireless LAN and Gigabit Ethernet LAN, respectively. And also, ensure that both GA server and GA mobile client were present in a same LAN. Initially ping test is done from the server to the client, the network delay is found to be 3ms. Dummynet, Wireshark and AssaultCube are installed in the GA server before performing experiments. The reason behind the selection of tools were explained in the following sections.

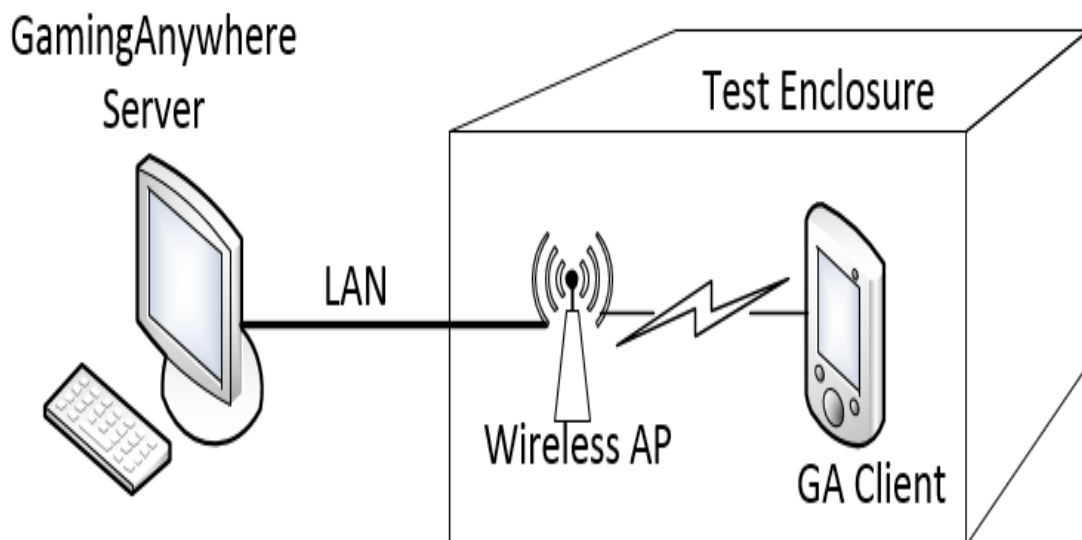


Figure 4.1: Experimental test-bed

The technical specifications of the components of the experimental test bed are shown in the following table

Devices	Technical Specification	Description
Server	Intel Quad Core processor, 4GB RAM, Windows 7	GA Version 0.8.0
Client	Samsung I9100 Galaxy SII, Dual-core 1.2 GHz Cortex-A9, , ~217 pixel density, Android v4.1	GAClient- v22
RF Shielded Test Enclosure	Ramsey STE3000FAV	It was used to avoid external signal interference.
WLAN	D-link DAP-1522, Firmware 1.21	Wireless Access point.

Table 4.1: Technical specifications

4.3.2 GamingAnywhere

GamingAnywhere includes the implementation of both server and client. To perform the experimentation with GamingAnywhere, the server and client's modules are launched along with them. As the advantage of being open-source, configuring it is easy when compared to other cloud gaming systems.

GA Server: When the GamingAnywhere is launched, its modules such as Audio source, Video source, RTSP server and input replayer were also launched. The input replayer and RTSP server waits for the incoming clients. If once the process was initialized, the video source and audio source were kept to idle. The encoder threads are launched when the client get connected to the RTSP server and then the encoder should send a notification to the respective module that it is waiting to encode the frames that are captured. Then the Video and audio frames are captured by the source modules when more than one encoder get to work. The encoded video and audio frames are generated all at once[19].

GA Client: GA client presents the real time game screens and dispatch them as encoded video and audio frames. The client consists of two threads one is used to render video and audio frames and the other is used to handle user inputs[19].

GamingAnywhere client on mobile is established by installing android application called GA client on mobile. Fig 4.2 expose the GA client software architecture[20]. The application is coded in java but few components are carried out as loadable shared objects in native C and C++.

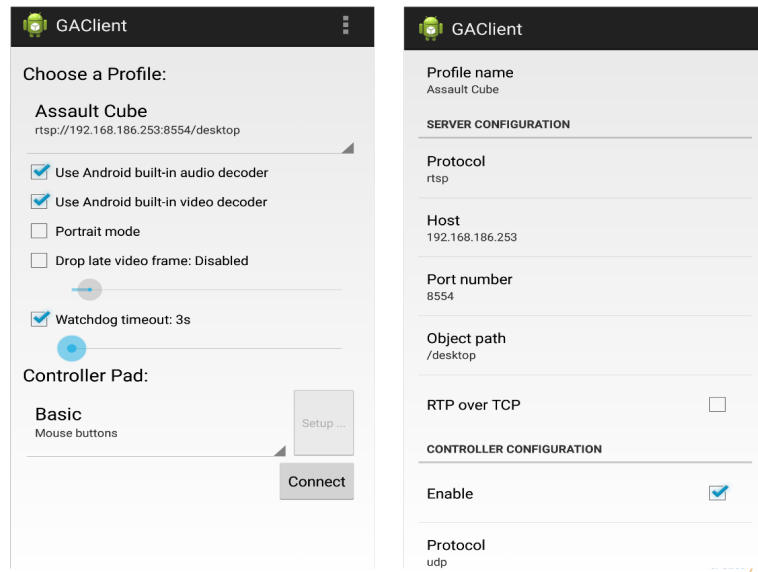


Figure 4.2: GA Client mobile application



Figure 4.3: Assault cube game running on mobile device

GamingAnywhere provides documentation to how to build server and client configuration files and also provided a bunch of configuration files which are already created. This study uses the Assaultcube configuration files given by GamingAnywhere with no alterations.

4.3.3 AssaultCube

AssaultCube game is selected in our research, since it is open-source, more realistic, free, multi-player and a first person shooter game. Moreover, the first person shooter games were more tempting, addictive, popular and well known genre[7]. The size of this game is 44MB only which is available in Windows, Linux and Macintosh platforms. The game is faster than Counter-Strike. There are a total of fourteen different game modes and two single-player bot game modes[1].

4.3.4 Test Enclosure

The RF shielded test enclosure, often called as Black box is used to avoid external interference in the network. The test enclosure used in our experiment is Ramsey STE3000FAV. Its is standard with 6 outlet 230 volt power strip. It has built-in high resolution camera that can record Audio and Video. It is provided with high efficiency RF gaskets and RF foam liner that helps in avoiding external interference.

In the thesis, the wireless AP and mobile device are enclosed in the test enclosure as shown in Fig 4.1. the mobile is placed inside the black box to isolate from outer signals.

4.3.5 Approach to the QoS

The research involves the measurement of QoS of mobile cloud gaming using GA in various scenarios. In order to effectively reach this goal, the recommendations by IETF is considered. The following are the two different QoS model approaches.

Integrated Service Model(IntServ): This model of approach guarantees the Quality of Service(QoS) with the focus on reservation of resources from the path of source to destination. Different protocols like RSVP, RTCP and RTP have been used by the applications to reserve the resources in the network. In this model, the process of request handling is developed to send the data over the network. In the research analysis of this thesis, the GA also uses RTP protocol for the video streaming[9].

Differentiated Service Model(DiffServ): This model is developed to overcome the problem of scalability in Integrated Service model. For the traffic flow in the network, this model has created different classes and the flows are aggregated into those classes. Each class receive a special treatment of QoS depending on the nature of the class. The different mechanism principles present in this architecture are traffic shaping, traffic conditioning, packet classifying, packet marking,

packet dropping etc. This model provides greater flexibility to employ the QoS in the network[8]. In this thesis, the research approach involves the measuring the above both QoS models.

4.3.5.1 QoS Metrics on survey

The performance metrics needed to measure the QoS of the cloud gaming are suggested by authors in [17].

Bandwidth: The total amount of network traffic transferred from source to destination within a particular interval of time. It is used to measure the capacity of the link between the two end points. It is measured in bits/sec.

Delay: It is defined as the time interval between the source sending the packet to the destination. It is also referred as one-way delay. If the time taken for destination to send the packet to the source is also considered, then it is referred as the round trip delay. It is measured as a unit of time. In our research, a delay of 33ms and 63ms are considered.

Jitter: It is defined as the difference in delay of received packets. It determines the consistency of connection between the source and the destination. It is also known as variation in end-to-end delay. It is measure as a unit of time. For continuous media transfers, this parameter must be taken into consideration when the system is not in deterministic state.

Packet Loss: It refers to the number of packets lost to reach the destination when being transmitted across the network. It is one of the important QoS parameter for real time application such as cloud gaming as of a case in our analysis. The packet loss values used in this research are 0.1% and 0.3%. In general, 0.1% packet loss means for every thousand packets transmitted there will be drop of one packet.

Bitrate: The number of bits i.e the amount of information transferred per unit of time. It is usually measured in bits per second (bps), kilo/mega bits per second (Kbps/Mbps).

Frame Rate: It is defined as the rate(frequency) at which the images(frames) are displayed. It is usually expressed in frames per second (fps). The frame rate used for this experiment is 50fps, since it is the default value set for the GA server.

Based on the survey of the research, the metric values were taken into the account as suggested by the authors Szigeti and Hettingh[30].As interactive video is related to cloud gaming, their recommendations were taken considered into the account for our experiments.

4.3.5.2 QoS Evaluation

To evaluate the QoS, certain performance evaluation tools are used in the research.

Dummynet: The network emulation is a technique useful in traffic shaping. Network simulation is different from the network emulation. This network emulation is used to alter the packet flow in a network. The traffic shapers are generally used to shape the network traffic by injecting traffic metrics like delay, bandwidth, packet loss etc. Dummynet[2] is a live network emulator used for bandwidth limitation, queuing/delay enforcement etc, It belongs to FreeBSD standard component and ipfw is its packet classifier. It runs on FreeBSD, Linux, OSX, Windows operating systems.

This has been chosen in the thesis for network emulation, to vary performance metrics like packet loss, delay and bandwidth limitation. Dummynet was installed and configured from the server. Dummynet works on the principle of creating pipes and passing packets to the pipes and implementing the certain rules. Rules are the ones which are configured and varied through given input parameters.

Wireshark: Wireshark[14] is free and open source network packet analyzer tool. This tool contains GUI used to analyze network protocols and gain insight into the network traffic. It is used to generate statistical network traffic data. In this research, the wireshark is used to measure bandwidth and protocol analysis.

Chapter 5

Experimentation and Results

5.1 Experimental Procedure



Figure 5.1: A Demonstration of experimental setup

The experimental set up consists of the GA server, GA client present in the RF test enclosure. The GA server runs the installed AssaultCube game and later the GA client is also set to run. A channel is created for the player at the GA client side. Once the streaming is started from the server to the client, the inputs from the mobile device on client side is sent to the server. This model is referred as Client-Server model.

Figure 5.1 shows the demonstration of experimental setup in our laboratory. Network traffic between the server and the client is controlled by the dummynet.

The performance metrics obtained in our research are bitrate, packet loss, jitter and frame rate. To calculate the metric values, a C++ programming script is written by the GamingAnywhere at the client side which generates a log file after the game session. This file is stored in the client side and later, a python script is written to retrieve the qos metric values. The different resolutions of the game used in the experiment are 1024x768, 800x600, 720x576, 640x480. These different resolutions of the game are tested in our experiments in different scenarios.

5.2 Trace Collection

Each game session lasted for 10 minutes. To collect the network traffic between the server and the client, Wireshark tool is used which captures the packets. Each game session traffic for different resolutions are stored and later analyzed to calculate the network bandwidth using wireshark which is shown in below Table 5.1. The formula used for bandwidth calculation is

$$\text{Bandwidth} = \left(\frac{\text{Total bytes of data sent} + \text{Total bytes of data received}}{\text{Total time duration}} \right)$$

Resolution	Bandwidth(Mbps)
1024x768	3.31
800x600	3.15
720x576	3.11
640x480	2.88

Table 5.1: Average bandwidth for different resolutions

5.3 Experimental Scenarios

Different experimental scenarios are carried out in the experimental research for 40 iterations.

5.3.1 Scenario 1

Resolution	Bitrate(Kbps)	PacketLoss(%)	Jitter(ms)	Frame Rate(fps)
1024x768	1400.523	0	689.97	44.2
800x600	1427.142	0	663.71	46.79
720x576	1494.825	0	716.4	47.92
640x480	1542.9	0	657.725	49.27

Table 5.2: QoS metrics for ideal case

In the first scenario, the performance metrics for the four different resolutions of the game are calculated without injecting the network traffic, i.e the ideal case scenario shown in Table 5.2.

5.3.2 Scenario 2

Resolution	Bitrate(Kbps)	PacketLoss(%)	Jitter(ms)	Frame Rate(fps)
1024x768	1353.9	9.37	661.47	22
800x600	1366.95	9.91	690.285	17.48
720x576	1380.425	10	751.45	16.87
640x480	1395.19	10.06	796.5	16.84

Table 5.3: QoS metrics for 0.1% packet loss

Resolution	Bitrate(Kbps)	PacketLoss(%)	Jitter(ms)	Frame Rate(fps)
1024x768	1047.125	29.61	680.325	1.9092
800x600	1057.16	29.74	683.19	1.805
720x576	1070	30.02	700.23	1.95
640x480	1080.375	30.47	708.55	1.81

Table 5.4: QoS metrics for 0.3% packet loss

In the second scenario, the packet loss of 0.1% and 0.3% are injected into the network traffic using dummynet. The performance metrics are evaluated in table 5.3 and 5.4 respectively.

5.3.3 Scenario 3

Resolution	Bitrate(Kbps)	PacketLoss(%)	Jitter(ms)	Frame Rate(fps)
1024x768	1476.2	0	674.95	45.9
800x600	1500.95	0	674.4	48.5
720x576	1530.075	0	606.625	49.38
640x480	1546.5	0	649.525	49.28

Table 5.5: QoS metrics for 33ms delay

Resolution	Bitrate(Kbps)	PacketLoss(%)	Jitter(ms)	Frame Rate(fps)
1024x768	1518.15	0	568.5	48.89
800x600	1534.35	0	694.55	48.78
720x576	1546.75	0	675.3	48.96
640x480	1550.475	0	670.9	49.17

Table 5.6: QoS metrics for 63ms delay

In the third scenario, the performance metrics are shown in Table 5.5 and 5.6 when the delay of 33ms and 63ms are injected into the network traffic respectively.

5.3.4 Scenario 4

Resolution	Bitrate(Kbps)	PacketLoss(%)	Jitter(ms)	Frame Rate(fps)
1024x768	991.75	20.16	700.7	4.82
800x600	993.35	21.3	700.2	5.06
720x576	994.875	21.58	710.9	3.06
640x480	995	21.8	718.025	3.67

Table 5.7: QoS metrics for bandwidth of 2000 Kbps

Resolution	Bitrate(Kbps)	PacketLoss(%)	Jitter(ms)	Frame Rate(fps)
1024x768	1194.65	12.56	688.05	9.93
800x600	1208	13.33	680.05	12.52
720x576	1203.05	12.18	745.3	11.71
640x480	1211.4	13.4	642.02	10.98

Table 5.8: QoS metrics for bandwidth of 2500 Kbps

In the fourth scenario, the performance metrics are shown in Table 5.7 and 5.8 when bandwidth is limited to 2000Kbps and 2500Kbps respectively.

Chapter 6

Analysis and Discussion

The aim of the thesis is to quantify the performance of QoS in mobile cloud gaming using GamingAnywhere. So, it is necessary to evaluate the quality of service under congested network. This chapter focuses on the analysis of the measured performance metrics in each scenario. The metrics bitrate, packet loss, jitter and frame rate have been measured for four resolutions of the game for 40 times in each 10 minutes scenario to determine the performance of mobile cloud gaming using GamingAnywhere. For better understanding, we considered only one game. In this chapter, section 6.1 describes the performance metrics such as bitrate while section 6.2 deals with packet loss. Finally section 6.3 and 6.4 focused on jitter and frame rate respectively.

6.1 Bitrate performance

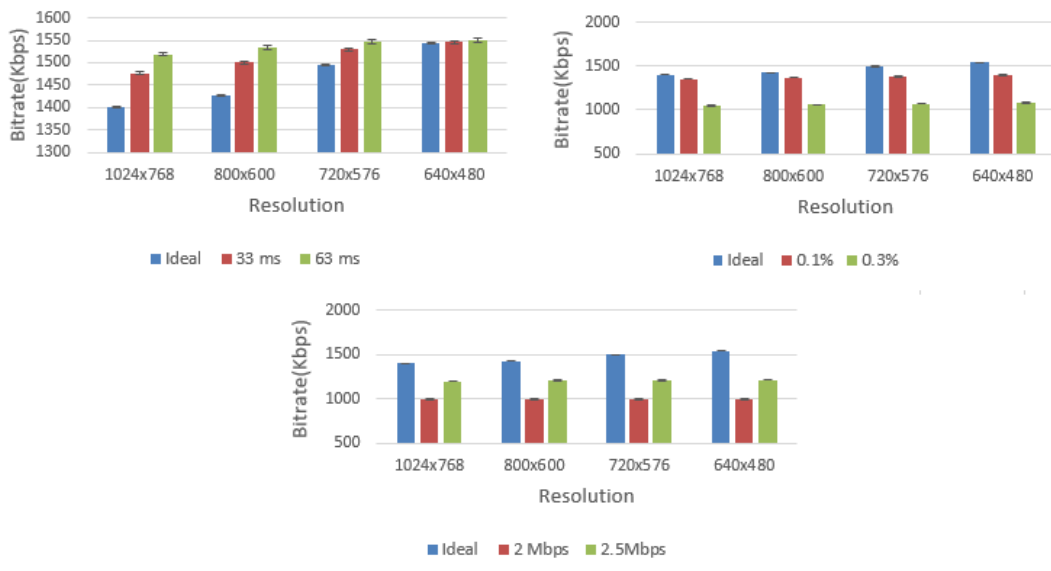


Figure 6.1: Performance of bitrate under different network conditions

This section gives the performance analysis of bitrate under different network conditions. These bitrate values are calculated when the RTP packets are transferred from GA server to the GA client. From figure 6.1, it indicates three graphs i.e bitrate affected by delay, bitrate affected by packet loss and bit rate affected by bandwidth limitation have been measured. The bar graph clearly shows that the bit rate value is increasing in the ideal case scenario with the decrease of resolution. Considering for each resolution, the bit rate value increases with the injection of network delay(33ms, 63ms). Similarly, the bit rate value increases with the injection of packet loss(0.1%,0.3%) and bandwidth limitation(2Mbps,2.5Mbps) when compared to ideal case scenario. The main objective of any software is to achieve minimum resource consumption in order to allow maximum resources to be available for the end users. Here bitrate is calculated with respect to time. So for the higher resolution, the data takes more time to transmit compared to lower resolution. The bitrate is higher for the lower resolution. The bar graph also depicts the higher bitrate for lower resolutions. With the inference of packet loss, delay and bandwidth, the number of packets will be dropped depending upon percent of loss injected and the latency respectively. From results, packet loss and bandwidth have imposed negative impact on the bitrate.

6.2 Packet Loss performance

In the server client model, packet loss is calculated by the following mathematical formula:

$$\text{Packet Loss} = \left(\frac{\text{Total packets sent} - \text{Total packets received}}{\text{Total packets sent}} \right) \times 100$$

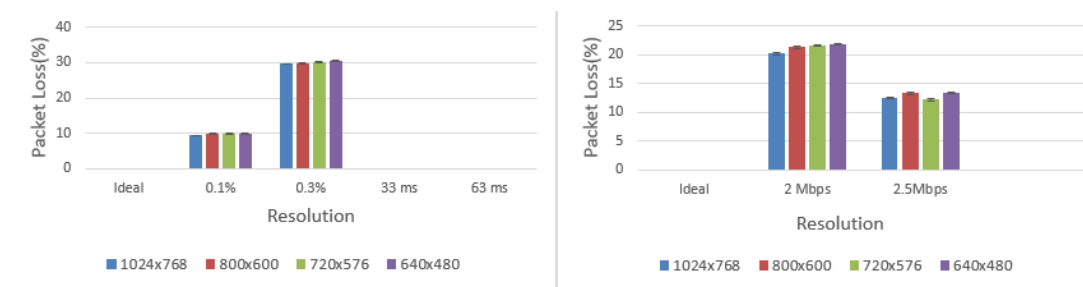


Figure 6.2: Performance of packet loss under different network conditions

This section analyses the performance of packet loss for each resolution of the game at GA client. Figure 6.2 depicts the performance of the packet loss in different network conditions. In this research, experiment results found that there is no single loss of packet in the GA mobile cloud gaming when streamed a game in any resolution. In general context, if the percentage of injected network

packet loss increases, the number of packets being lost in the network will also increase. Figure 6.2 also shows that the GA mobile client's packet loss increased when the injected network packet loss increase. Also, the packet loss in different resolutions are almost the same having slight changes which can't be considered into account. In the limitation of bandwidth scenario, the transfer of packets will be dropped due to the arise of network congestion problem.

6.3 Jitter performance

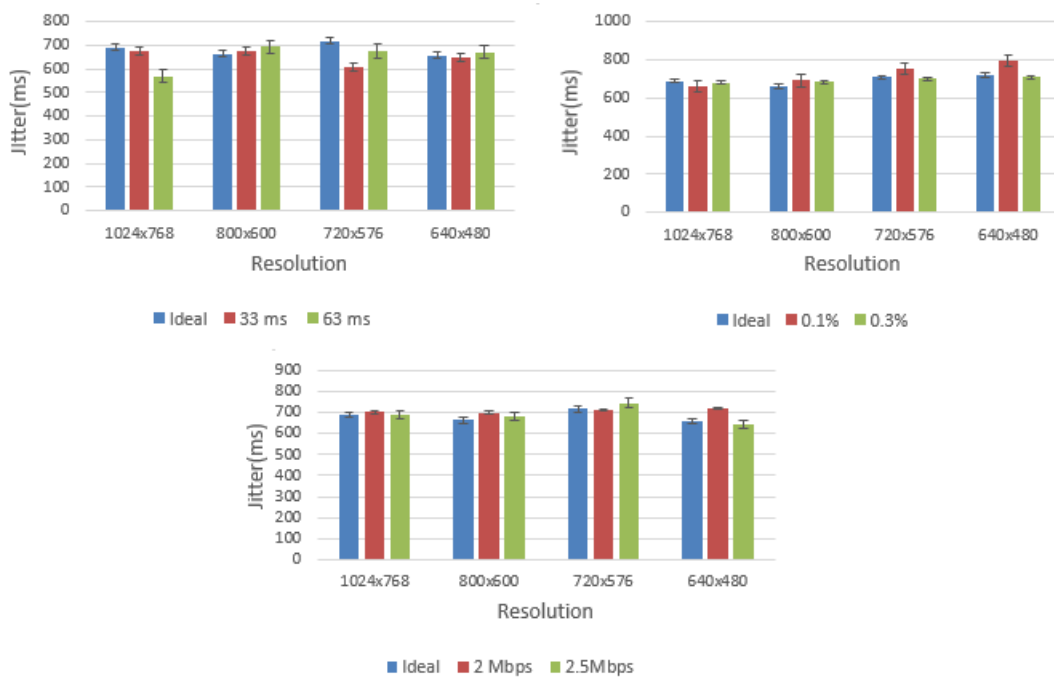


Figure 6.3: Performance of jitter under different network conditions

This section explains about the performance of jitter in different network conditions. From the Figure 6.3, if we observe all graphs carefully, the jitter values are slightly varying with different resolutions in different scenarios.

After analysing the GA code, Jitter is calculated in timestamp units. In the GA mobile cloud gaming, an audio stream is sampled at frequency of 90000Hz, the inter arrival time measured with the local clock is converted by multiplying the seconds by 90000. For several RTP packets within a video frame have the same time stamp, the first packet in the frame is advised to compute the jitter.

And also sometimes, the noise is added into the jitter calculation if the encoders that transmit data in a different order than it was actually sampled. Corruption of synchronization signals or electromagnetic interference during video transmission also effects the jitter.

6.4 Frame Rate performance

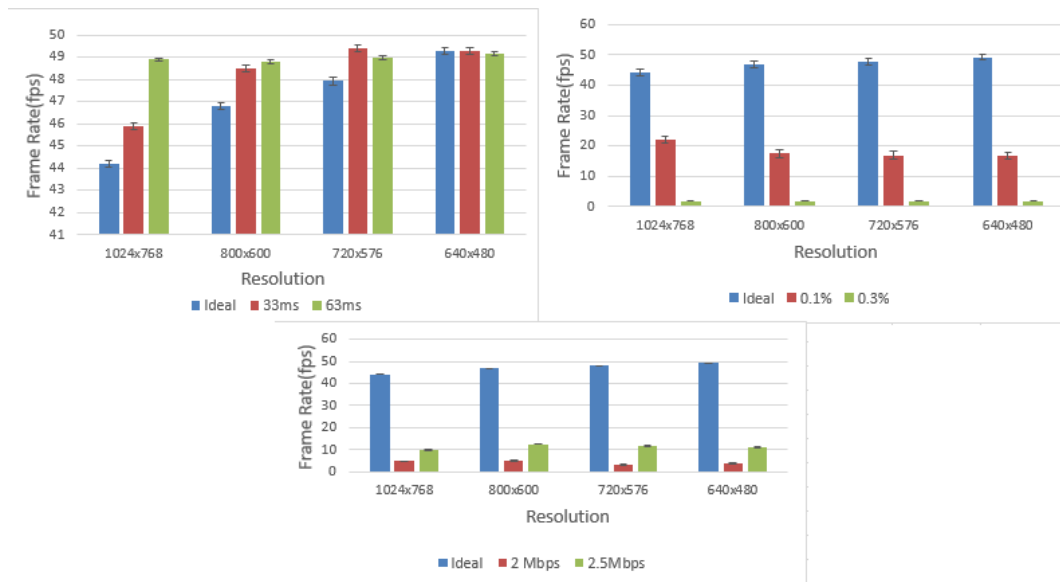


Figure 6.4: Performance of frame rate under different network conditions

This section describes about the frame rate performance in mobile cloud gaming using GamingAnywhere. Before injecting network traffic, frame rate is around 49 fps at the client side. In GA, the frame rate used at the H264- encoder is 50 fps. In this research, the default setting used is 50fps as suggested by GamingAnywhere. Injecting the packet loss and bandwidth limitation has imposed a negative impact on frame rates of GA. Frame rate is directly proportional to bitrate. Lower the bitrate is to lower the framerate. The bar shown in the figure decreases when injected network packet loss increases.

Chapter 7

Conclusions and Future Work

This chapter outlines the conclusion from the research. It provides a section for the possible future work that can be conducted to contribute to the field of study of mobile cloud gaming and improve the study of software switch performance. It answers the previously mentioned research questions individually.

7.1 Conclusion

In this research, we have seen how to quantify the performance of QoS of mobile cloud gaming using GamingAnywhere. Primarily, the game AssaultCube is streamed from GamingAnywhere server to the GamingAnywhere mobile client and the QoS metrics is measured in terms of bitrate, packet loss, jitter and frame rate. And also, the QoS metrics are measured under diverse network conditions using the traffic shaper Dummynet. The results obtained from QoS metrics have given an insight into the performance of mobile cloud gaming using GamingAnywhere in different scenarios.

From the analysis of the statistical results, as resolution varied from 1024px to 640px, the performance metrics for GamingAnywhere mobile cloud gaming i.e the bitrate and frame rate are higher, packet loss is zero, jitter has decreased. When injected to the network conditions, jitter also slightly increases with respect to the network packet loss and bandwidth limitation. Packet loss and bandwidth limitation have imposed a negative impact on the frame rate. In the research we have achieved a higher frame rate and higher bitrate with the impact of network delay.

ANSWERS TO THE RESEARCH QUESTIONS

1. **How much network bandwidth is utilized by a game session?** In this experiment, Wireshark networking tool is used to calculate the network bandwidth. This question can be answered by observing the table 5.1. The network bandwidth is shown for each different resolutions of the game. Overall considering the average network bandwidth consumed in a game

session is 3.11 Mbps. The calculation for the network bandwidth is shown in the section 5.2.

2. How can we achieve trade-off for QoS for different screen resolutions with the impact of network delay?

Ans. This question can be answered by observing the experimental results of scenarios 1 and 3 in the section 5.3. Bitrate performance is increased with the decrease of screen resolutions when 33ms, 63ms delay are injected. Packet loss performance is constant i.e zero with or without the impact of network delay. The value of jitter slightly decreases with the decrease of screen resolution with the impact of network delay. Also, the value of frame rate has been increased with the impact of network delay.

3. What is the impact on QoS metrics when subjected to variable packet loss in the network?

Ans. In the experiment, the traffic shaper is used to inject variable packet losses in the network to analyze the performance of GamingAnywhere. The table 5.2, 5.3 and 5.4 shows the values of QoS metrics like bitrate, packet loss, jitter and frame rate. The statistical view for the performance of QoS metrics have been shown in the different sections of the sixth chapter. Packet loss has imposed a negative impact on the QoS metrics performance of the GamingAnywhere. Also, the analysis of the performance of QoS metrics has been explained in the sections of the sixth chapter which gives detailed explanation to this question.

4. What is the impact on QoS metrics when the network bandwidth is limited?

Ans. Similarly as mentioned above, the dummynet traffic shaper is used for the limitation of the network bandwidth. The table 5.2, 5.7 and 5.8 shows the values of QoS metrics. Bitrate, frame rate are decreased with the limitation of bandwidth. The value of jitter and packet loss are also increased with this impact. Also, the statistical comparison analysis explained in different sections of chapter 6 gives the detail explanation with impact of network bandwidth. Atlast, the performance of GA mobile cloud gaming degrades with the impact of network bandwidth.

7.2 Future Work

This thesis is focused on the smaller time-scale performance metrics of mobile cloud gaming for a single FPS game. Quantifying performance metrics for larger time-scale metrics and with the different strategical and action adventurous games can be extended as a future work. Since, our experimentation is conducted over

Windows platform, other platforms like Linux, OS X can also be done as a future work.

References

- [1] Assault cube game. <http://assault.cubers.net>. Accessed: 2016-09-14.
- [2] Dummynet. <http://info.iet.unipi.it/~luigi/dummynet/>. Accessed: 2016-09-14.
- [3] Gaikai cloud gaming system. <https://www.gaikai.com/>. Accessed: 2016-09-10.
- [4] GamingAnywhere Website. <http://gaminganywhere.org/>. Accessed: 2016-09-10.
- [5] OnLive cloud gaming system. <http://onlive.com/>. Accessed: 2016-09-10.
- [6] StreamMyGame cloud gaming system. <http://streammygame.com/smg/index.php>. Accessed: 2016-09-10.
- [7] Thomas H Apperley. Genre and game studies: Toward a critical approach to video game genres. *Simulation & Gaming*, 37(1):6–23, 2006.
- [8] Steven Blake, David Black, Mark Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss. An architecture for differentiated services. Technical report, 1998.
- [9] Robert Braden, David Clark, and Scott Shenker. Integrated services in the internet architecture: an overview. Technical report, 1994.
- [10] W. Cai, V. C. M. Leung, and M. Chen. Next generation mobile cloud gaming. In *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*, pages 551–560, March 2013.
- [11] K. T. Chen, Y. C. Chang, H. J. Hsu, D. Y. Chen, C. Y. Huang, and C. H. Hsu. On the quality of service of cloud gaming systems. *IEEE Transactions on Multimedia*, 16(2):480–495, Feb 2014.
- [12] Kuan-Ta Chen, Yu-Chun Chang, Po-Han Tseng, Chun-Ying Huang, and Chin-Laung Lei. Measuring the latency of cloud gaming systems. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1269–1272. ACM, 2011.

- [13] S. P. Chuah, C. Yuen, and N. M. Cheung. Cloud gaming: a green solution to massive multiplayer online games. *IEEE Wireless Communications*, 21(4):78–87, August 2014.
- [14] Gerald Combs et al. Wireshark-network protocol analyzer. *Version 0.99*, 5, 2008.
- [15] John W Creswell. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2013.
- [16] M Fiedler et al. State-of-the-art with regards to user-perceived quality of service and quality feedback. *Information Society Technologies (IST) Sixth Framework Programme 2004*, 2004.
- [17] Tristan Henderson and Saleem Bhatti. Networked games: a qos-sensitive application for qos-insensitive users? In *Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS: What have we learned, why do we care?*, pages 141–147. Acn, 2003.
- [18] Chun-Ying Huang, De-Yu Chen, Cheng-Hsin Hsu, and Kuan-Ta Chen. Gaminganywhere: An open-source cloud gaming testbed. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, pages 827–830, New York, NY, USA, 2013. ACM.
- [19] Chun-Ying Huang, Cheng-Hsin Hsu, Yu-Chun Chang, and Kuan-Ta Chen. Gaminganywhere: An open cloud gaming system. In *Proceedings of the 4th ACM Multimedia Systems Conference*, MMSys '13, pages 36–47, New York, NY, USA, 2013. ACM.
- [20] Chun-Ying Huang, Cheng-Hsin Hsu, De-Yu Chen, and Kuan-Ta Chen. Quantifying user satisfaction in mobile cloud games. In *Proceedings of Workshop on Mobile Video Delivery*, MoViD'14, pages 4:1–4:6, New York, NY, USA, 2013. ACM.
- [21] TSS ITU. Terms and definitions related to quality of service and network performance including dependability. *Recommendation E*, 800, 1994.
- [22] Michael Jarschel, Daniel Schlosser, Sven Scheuring, and Tobias Hoßfeld. An evaluation of qoe in cloud gaming based on subjective tests. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*, pages 330–335. IEEE, 2011.
- [23] Miika Komu, Roberto Morabito, Tero Kauppinen, Jimmy Kjällman, and Yong Yao. Power consumption in remote gaming: an empirical evaluation.

- [24] Ulrich Lampe, Ronny Hans, and Ralf Steinmetz. Will mobile cloud gaming work? findings on latency, energy, and cost. In *Proc. of IEEE International Conference on Mobile Services (MS'13)*, pages 960–961, 2013.
- [25] Averill M Law and W David Kelton. Building valid, credible, and appropriately detailed simulation models. *Simulation Modeling and Analysis*, 3:264–291, 2000.
- [26] P. E. Ross. Cloud computing’s killer app: Gaming. *IEEE Spectrum*, 46(3):14–14, March 2009.
- [27] Andreas Sackl, Raimund Schatz, Tobias Hossfeld, Florian Metzger, David Lister, and Ralf Irmer. Qoe management made uneasy: The case of cloud gaming. In *2016 IEEE International Conference on Communications Workshops (ICC)*, pages 492–497. IEEE, 2016.
- [28] R. Shea, J. Liu, E. C. H. Ngai, and Y. Cui. Cloud gaming: architecture and performance. *IEEE Network*, 27(4):16–21, July 2013.
- [29] Omar Soliman, Abdelmounaam Rezgui, Hamdy Soliman, and Najib Manea. *Mobile Cloud Gaming: Issues and Challenges*, pages 121–128. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [30] Tim Szigeti, Christina Hattingh, Robert Barton, and Kenneth Briley Jr. *End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks*. Cisco Press, 2013.
- [31] Zhihong Zhuo. A performance comparison of vmware gpu virtualization techniques in cloud gaming. 2016.

Appendix A

GamingAnywhere Scripts

SERVER SCRIPTS

ga-server-event-driven config/server.assaultcube.win32.conf

CLIENT SCRIPTS

ga-client config/client.rel.conf rtsp://192.168.186.244:8554/desktop

AssaultCube Configuration File

```
#configuration for the assaultcube game #work with ga-server-event-driven [core]
include=common/server-common.conf include=common/controller.conf include=
common/video-x264.conf include=common/video-x264-param.conf include=common
/audio-lame.conf [video] video-fps=50[audio] audio-init-delay=1000 [filter] filter-
source-pixelformat=rgba [ga-server-event-driven] game-dir=D:\AssaultCubePortable
\App \assaultcube game-exe=bin_win32 \ac_client.exe # hook configuration
# version: d9, d10, d10.1, d11, dxgi, sdl hook-type=sdl hook-audio=coreaudio
enable-audio=true enable-server-rate-control=Y server-token-fill-interval=20000
server-num-token-to-fill=1 server-max-tokens=2
```

Appendix B

Dummynet Rules

```
#ipfw -q flush  
#ipfw -q pipe flush  
#ipfw add <number> pipe <number> ip from any to any  
#ipfw pipe <number> config delay <value>  
#ipfw pipe <number> config plr <value>  
#ipfw pipe <number> config bw <value>
```

Rule and pipe numbers are changed in each experiment and values given are 33ms, 63ms for delay and 0.1, 0.3 for packet-loss and also 2048Kbit/s, 2500Kbit/s for bandwidth limitation respectively.