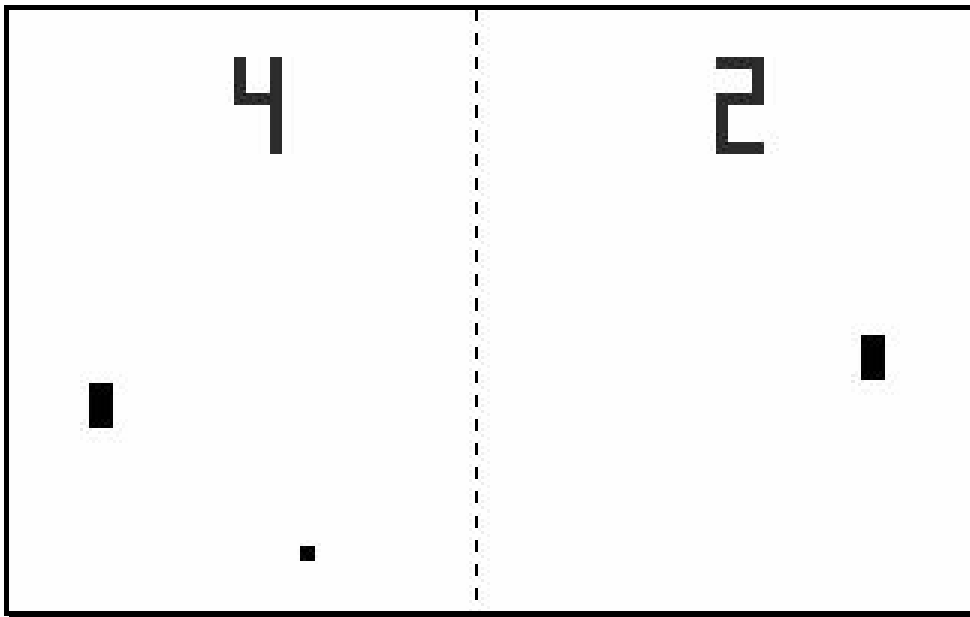


Performance Modeling Of Interactive Gaming



A.F. Wattimena
May 2006



Performance Modeling of Interactive Gaming

Alexander Franciscus Wattimena



University:
Vrije Universiteit
Faculty of Sciences
Business Mathematics and Informatics
De Boelelaan 1081a
1081 HV Amsterdam

Supervisors:
Sandjai Bhulai
Rob van der Mei

Company:
TNO Informatie- en Communicatietechnologie
Brassersplein 2
Postbus 5050
2600 GB Delft

Supervisors:
Rob Kooij
Hans van den Berg



Oct 2005 - May 2006

Executive Summary

Interactive gaming has taken an enormous flight since the Internet enabled gamers to play over a globally accessible network. Increasingly, gaming looks to be one of the mainstay applications that help to strengthen retention and loyalty of high-speed Internet consumers. By providing a superior gaming experience that produces reliable performance, broadband operators can likely retain relationships for years with the next generations of customers. However, in the past online gaming has been mainly approached from a marketing point of view. Relatively little time has been spent on scientific research. As a consequence, large gaming service providers often lack information on both gamers' demands and estimations of delay times (ping times). This type of information is crucial before deploying large-scale real-time services over the Internet.

The objective of this research project was to provide insight into the estimation of ping times and into the quality experience of gamers.

In "Part I: Gaming Performance Model" the influence of different network parameters on First Person Shooter ping times was studied. We followed a mathematical approach which heavily relied on queuing theory. The ultimate result was a Mathcad implementation which enabled us to calculate delay quantiles for different network scenarios. We were particularly interested in the influence of VoIP traffic running over the same network. We found out that the addition of voice packets above 25% of total traffic leads to a decrease in ping times, while smaller fractions of additional voice packets result in an increase in ping times (compared to the situation in which only gaming traffic is running over the network). Other parameters of interest were the load on the aggregation link, the inter-arrival time of gaming bursts and the Erlang order K of mean burst sizes. The first two mentioned parameters have a negative impact on the network quality of the FPS gamers: ping quantiles are significantly higher for increased parameter values. Increasing the Erlang order K has the opposite effect: the decrease in variation of the mean burst size results in less congestion in the system.

In "Part II: G-model" we focused on the influence of network impairment on subjective and objective performance measures. We conducted a user experiment to quantify this impact. Our experiment results demonstrate that ping and jitter have a significant negative effect on both the subjective and objective gaming quality, while packet loss goes unnoticed for values up to 40%. Making use of a multi-dimensional regression analysis we developed the Quake IV G-model which enabled us to predict a gamer's quality rating (expressed in a Mean Opinion Score) based on ping and jitter values. A follow-up validation experiment showed that the model is very accurate in estimating MOS values. A similar regression model was constructed to relate the network impairment to the objective performance measure kills. Concerning the objective performance of gamers, we conclude that although a player's score is significantly affected by ping and jitter, interpretation of an objective measure value is very hard because of the additional dependence on other factors like the number of players, the size of the map and the skill level of the players.

Preface

This report is the final product of my internship at TNO Information and Communication technology. The internship is an obligatory part of the Master program of Business Mathematics and Informatics. From October 2005 to May 2006 I have conducted my research on performance aspects of interactive gaming as a trainee within the department Planning, Performance, and Quality.

I would like to take this opportunity to thank my TNO supervisor Rob Kooij for his great contribution to my graduation project. I have really enjoyed our cooperation, which was not bounded by the walls of the TNO office. His sparkling personality had a positive influence on both my project as well as on me as a person. Besides his sharp insights and remarks I am very grateful for all the opportunities he has created for me during my time at TNO. Also, I would like to thank my VU supervisors Sandjai Bhulai and Rob van der Mei for their valuable feedback.

Furthermore, I would like to thank Kamal Ahmed, Jeroen van Vugt, John Beerends and Rob Blokland for their support during the setup and execution of our gaming experiment. Of course my gratitude goes out to the participating players as well. I would specially like to thank Nono van den Hoek Ostende a.k.a. Tex, who hooked me up with some of Holland's finest Quake IV players.

Altogether, the seven months at TNO have been great and would not have been the same without the fantastic group of trainees.

I hope you enjoy reading this report as much as I enjoyed writing it.

Frank, May 2006

Table of contents

EXECUTIVE SUMMARY	V
PREFACE	VII
TABLE OF CONTENTS	IX
1 INTRODUCTION	11
1.1 PROBLEM DESCRIPTION	11
1.2 OVERVIEW.....	12
2 TNO	13
2.1 TNO INFORMATION AND COMMUNICATION TECHNOLOGY	14
2.2 DEPARTMENT PLANNING, PERFORMANCE & QUALITY	14
3 GAMING OVERVIEW	15
3.1 MULTIPLAYER GENRES	15
3.2 INDUSTRY ANALYSIS.....	16
4 PART I: GAMING PERFORMANCE MODEL	19
4.1 INTRODUCTION.....	19
4.2 TRAFFIC SOURCE MODEL.....	20
4.3 UPSTREAM	21
4.4 DOWNSTREAM.....	23
4.4.1 <i>Delay distribution of batches</i>	24
4.4.2 <i>Delay distribution of packets</i>	29
4.5 TOTAL DELAY.....	30
4.6 RESULTS	31
5 PART II: G-MODEL	37
5.1 INTRODUCTION.....	37
5.2 RELATED WORK	37
5.3 FACTORS THAT INFLUENCE THE PERCEIVED QUALITY OF GAMERS	38
5.4 THE EXPERIMENT	40
5.4.1 <i>Setup</i>	42
5.4.2 <i>Questionnaires</i>	43
5.5 RESULTS	44
5.5.1 <i>Factor analysis</i>	44
5.5.2 <i>Regression Analysis</i>	44
5.5.3 <i>G-model</i>	44
6 CONCLUSIONS AND RECOMMENDATIONS	45
6.1 CONCLUSIONS	45
6.2 RECOMMENDATIONS.....	46
REFERENCES	47
APPENDIX A: MATHCAD CALCULATION SHEETS	49
APPENDIX B: EXPERIMENT SCENARIOS	52
APPENDIX C: ADDITIONAL GRAPHS	53

1 Introduction

Interactive gaming has taken an enormous flight since the Internet enabled gamers to play over a globally accessible network. According to a report on backbone traffic 6 popular games make up for three to four percent of the traffic already [7]. Such numbers are expected to keep rising dramatically; percentages of up to 25% of total backbone traffic have been predicted for the year 2008. A number of factors contribute to this explosive growth of online gaming, of which the ongoing penetration of broadband connections and the introduction of console platforms are only two [18]. Increasingly, gaming looks to be one of the mainstay applications that help to strengthen retention and loyalty of high-speed Internet consumers. By providing a superior gaming experience that produces reliable performance, broadband operators can likely retain relationships for years with the next generations of customers [21].

1.1 Problem description

In the past online gaming has been mainly approached from a marketing point of view. Relatively little time has been spent on scientific research. This is possibly a consequence of the best effort character of the Internet. This best effort approach appears to lose ground to a Quality of Service approach, similar to developments in the VoIP market. Large gaming service providers lack information on both gamers' demands and estimations of delay times (often denoted as ping times). This type of information is crucial before deploying large-scale real-time services over the Internet.

The goal of this project is two-fold:

To provide insight into the estimation of ping times and into the quality experience of gamers

Concerning the ping times the following questions will be answered:

- In what way does interfering voice traffic influence ping times of online gamers?
- What is the influence of different network parameters on estimated ping times?

Regarding the quality experience the following issues will be addressed:

- Which factors are responsible for influencing the quality experience of online gamers?
- What is the relation between the perceived quality of gamers and these factors?
- Can we identify a relation between an objective performance measure and these factors?

In order to provide the desired insights two models will be developed:

I Gaming Performance Model

II G-model

The Gaming Performance Model is an extension to the queuing model described in [5]. The extension will be able to quantify the impact of VoIP traffic running over the same network.

The G-model, analogous to the E-model for VoIP [1, 15], will be designed to capture the user's quality experience in a Mean Opinion Score.

1.2 Overview

This report starts with a description of TNO and the department where this research was carried out (Chapter 2). Subsequently, an overview of the gaming industry will be given in Chapter 3. The heart of the report can be roughly divided into two sections:

In Chapter 4 the Gaming Performance Model is constructed which will follow a theoretical approach for assessing the delay performance of a typical interactive game. First the problem background will be discussed, including a description of the underlying traffic source model which describes the main traffic characteristics of modern First Person Shooters. The queuing model is divided into an upstream and downstream component. In Section 4.4 the two separate parts are combined to calculate the total delay in the queuing system. The chapter will be concluded with an overview of numerical results.

In Chapter 5 the previously mentioned G-model is constructed. The chapter starts with an overview of existing literature on the subjective quality experience of gamers. In Section 5.3 the final set of experiment factors is discussed, before moving over to Section 5.4 which focuses on the gaming experiment setup. The results in Section 5.5 give an outline of the relations between the chosen network parameters and both the objective and subjective measures.

Chapter 6 reviews the most important results of this research project, along with a number of recommendations for future research.

This report is concluded with the reference section and the appendices.

2 TNO

Mission statement:

TNO makes scientific knowledge applicable in order to strengthen the innovative capacity of business and government [28]

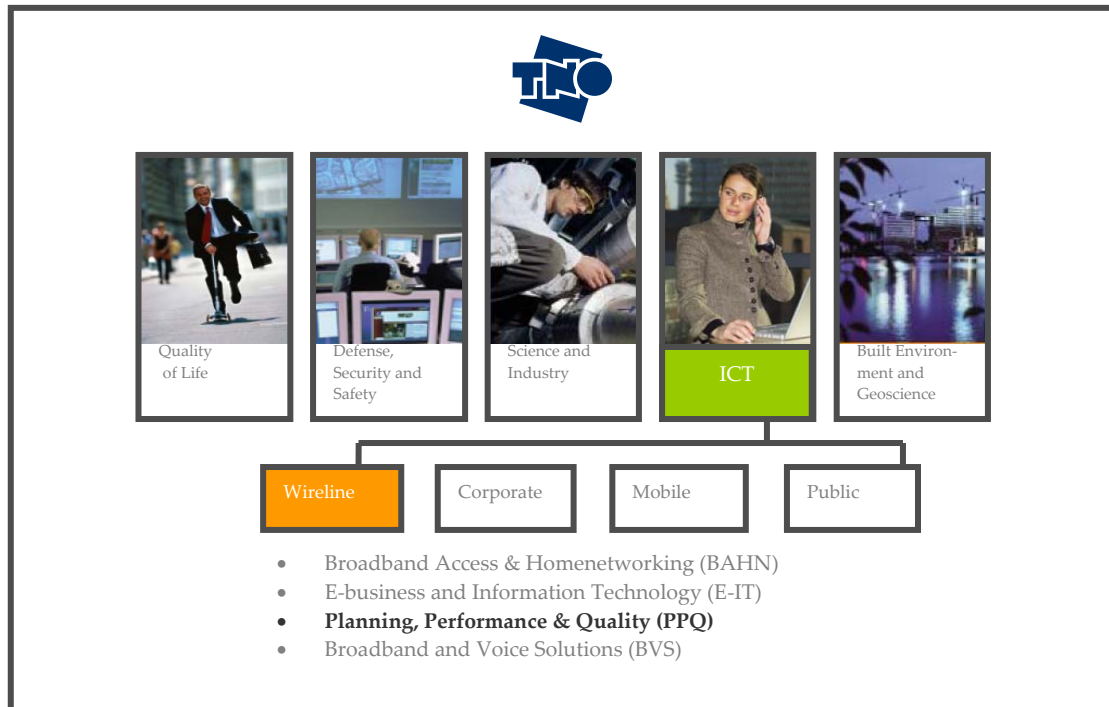


Figure 1: TNO organization scheme

TNO is a knowledge organization for companies, government bodies and public organizations. The daily work of some 5,000 employees is to develop and apply knowledge. The company provides contract research and specialist consultancy as well as grant licenses for patents and specialist software. They test and certify products and services, and issue an independent evaluation of quality. And TNO sets up new companies to market innovations. The development and application of innovative knowledge: that is what TNO is all about.

2.1 TNO Information and Communication Technology

TNO Information and Communication Technology is a unique centre of innovation in the Netherlands that unites the ICT and Telecom disciplines of TNO. TNO ICT helps companies, government bodies and (semi-) public organizations to realize successful innovations in ICT. Value creation for clients is the priority, and the added value lies in the combination of innovative strength and in-depth knowledge. The approach to innovation is integrated and practical. Research involves more than the technologies themselves. Where necessary, TNO ICT also focuses on user-friendliness, financial aspects, and business processes. Carrying out technical and market trials support the implementation process. They are also specialists in innovation strategy and policy, and the extensive ICT expertise is a valuable resource that can be used to address issues in the wider community.

2.2 Department Planning, Performance & Quality

The focus of Planning, Performance & Quality is to optimize the balance between costs and quality of ICT networks and services. In this way customers are able to guarantee quality requirements in a cost effective way. The expertise includes the development and application of models centered on performance analysis and network planning. It is based on a unique combination of both theoretical and practical knowledge of operations research and ICT networks. Developed products enable the customer to visualize decisions concerning design, parameter tuning and operational strategies. As a result decision-making is optimally supported. The department builds on a broad background of expertise which makes it possible to flexibly adapt to new technological developments within the market.



3 Gaming overview

“Games are evolving to entertain, educate, and engage us individually. These personalized games will reflect who we are and what we enjoy, much as our choice of books and music does now. They will allow us to express ourselves, meet others, and create things that we can only dimly imagine. They will enable us to share and combine these creations, to build vast playgrounds. And more than ever, games will be a visible, external amplification of the human imagination.” [31]

Computer games have been part of people’s lives for a fair amount of time now. Ever since Pong was released, many players around the world have enjoyed playing interactive games. Over time faster computers enabled game developers to enhance graphics and gameplay features. Many different games and gaming platforms (dedicated gaming consoles like Nintendo and Playstation, and multifunctional platforms like the Personal Computer and the Mac by Apple) were released over the years, but it was only until the nineteen nineties that the Internet enabled gamers to meet and play over a globally accessible network. Until that moment gamers were restricted to play against the computer or against friends at the same location (either on the same computer or over a Local Area Network). Clearly, this technological breakthrough opened up a whole new variety of options to both software- and hardware developers. Many different genres emerged within these interactive multiplayer games.

3.1 Multiplayer genres

The large portion of popular multiplayer games can be categorized using one of the following labels [30]:

- **First-person shooter**
A first-person shooter (FPS) is a combat computer or video game genre in which the user has a first person perspective of a relatively small environment. The main objective is to kill other players as often as possible, although some popular modifications like Capture The Flag have different goals.
Popular examples: Counter-strike (Valve), Quake IV (ID Software), Battle Field 2 (EA)
- **Real-time strategy**
A real-time strategy (RTS) game is a type of computer strategy game which does not have "turns" like conventional turn-based strategy video or board games. Rather, game time progresses in "real time": that is, it is continuous rather than turn-by-turn. (The word "strategy" originally referred to war planning at the level of armies and squadrons instead of giving orders for individual units or persons).
Popular examples: Starcraft, Warcraft III (both Blizzard), Age of Empires III (Microsoft)
- **MMORPG**
Massively Multiplayer Online Role-Playing Games (MMORPGs) follow a client-server model in which players, running the client software, are represented in the game world by an avatar — a graphical representation of the character they play. Providers (usually the game's publisher) host the persistent worlds these players inhabit. This interaction between a virtual world, always available for play, and an ever-changing, world-wide stream of players characterizes the Massively Multiplayer Online Role-Playing Game.
Popular examples: Everquest 2 (Sony), World of Warcraft (Blizzard)



- **Browser games**

Browser games are distinct from other video and computer games in the sense that they are normally platform independent, relying solely on client-side technologies (normally called 'applets' or 'plug-ins'). Usually all that is required to play browser games is an Internet connection, a web browser and the appropriate plug-ins (normally available for free via the plug-in maker's website).

Popular examples: Online Pictionary, Chess

- **Online gambling games**

Online gambling games are the virtual variants of betting games which are usually played in a casino. Just like regular casino games, the online versions are centered on money and winning (in most cases) requires a certain amount of luck. The online casino is usually a web site which hosts the different chance games and provides deposit and withdrawal functionality.

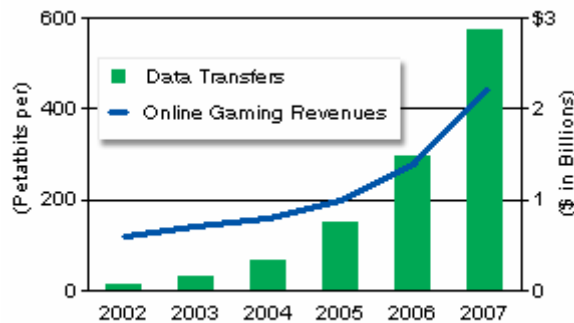
Popular examples: Blackjack, Poker

Next to the previously mentioned multiplayer categories, a number of single player genres are becoming more and more popular in a multi-user setting. Especially racing games like Need for Speed and sports games like FIFA Soccer are nowadays played on the Internet.

3.2 Industry analysis

The (online) gaming industry is booming and expected to keep growing explosively the next couple of years. Let us take a look at some numbers [21]:

- In 2003, total industry revenues rose to a record \$7 billion, making video games a bigger industry than the Hollywood movie business
- The online gaming market is expected to grow from about \$600 million in 2003 to over \$2 billion in revenues in 2007 (Figure 2)
- Today, about one third of new video games sold feature some sort of online component



Source: In-Stat/MDR 3/03

Figure 2: Total worldwide revenues and data transfers for online gaming [24]

A number of factors can be attributed to explain for the continuing growth of the online industry [18]:

- **Ongoing broadband penetration**

The large scale adaptation of high-speed Internet through Cable and DSL has a dramatic impact on the accessibility of online gaming. A high speed connection is a necessity for any gamer wanting to play over the Internet.

- **Adoption of more robust wireless networks**

Wireless next-generation networks are rapidly being deployed around the world, thereby enabling gamers to engage in online combat through their last generation handheld (Nintendo DS or Sony



PSP) or mobile phone. Mobile gaming already is a huge success in Japan, where gaming is responsible for 15% of iMode revenue.

- **Expansion of online activities by console platform manufacturers**

Online gaming roughly started halfway the nineties with the PC game DOOM. It was only until 2002 that two major console manufacturers Sony (Playstation) and Microsoft (Xbox) entered the online market with the release of Internet adapters. Writing this, Nintendo's Gamecube still lacks any decent support for online facilities. Trying to make up for this late introduction, the latest generation consoles from all three hardware manufacturers offer or promise (the Playstation 3 and Nintendo Revolution have not been released yet) extensive online capabilities (e.g., Xbox 360 with Xbox Live!).

- **Legislation of online gambling**

Online gambling is already legal in a big part of the world. It is also expected that countries with more conservative gambling policies will loosen their policies in the near future, because the missed revenue is too great to pass up. Online gambling enables people to enter (virtual) casinos without leaving their homes, thereby taking away barriers to engage in gambling activities. Especially poker has taken an enormous flight. Over \$180m is being wagered in cash game pots in online poker every day (April 2005), in comparison to just \$10m in 2003.

- **Maturing of the consumer base**

When talking about gamers, people tend to think of pimply male teens. And for many years this image has correctly covered the dominant part of the gaming community. But the last couple of years the consumer base seems to be shifting towards a more mixed audience; young and old, including a growing number of female gamers, meeting each other on the Internet to play computer games in an interactive setting (see Figure 3). This shift is mainly brought about by the explosive growth of Internet gambling (especially poker), the rise of casual gaming [22] and the releases of several MMORPG monster hits like Blizzard's 'World of Warcraft' and Sony's 'Everquest 2'. The MMORPG seems to attract a broader audience thanks to the creation of a virtual world in which gameplay is closely related to people's daily routine, in comparison to classic multiplayer genres like FPS and RTS which are extremely combat oriented.



Figure 3: Age and gender of American gamers [27]

- **Increasing media attention**

The explosion of the industry's popularity has not gone unnoticed. Internet journalists have written about the development of the industry since day one. But today also traditional media, like television and newspapers, are keeping a close look on what is going on in gaming land. Especially professional gaming events like the World Cyber Games [32] and The Cyberathlete Professional League [26] are getting a lot more coverage through traditional channels than before (e.g. [34]).

- **The rise of the cyber-athlete**

Korea is seen by many as the cradle of professional gaming, with hundreds of Starcraft and Warcraft III players under contract by large gaming clans. Important matches including commentary are broadcasted on three national channels. Although nowhere near the celebrity status of Korean pro-gamers, cyber-athletes from the United States and Europe are getting their



share of media attention [34]. This media coverage attracts more sponsors and thereby boosts the expansion of Electronic-sports in its turn. Following their Asian counterparts, gamers and other people from the community gather in national E-sports associations and by doing so get recognition from authorities just like regular sport associations.

- **In-game advertising and Advergaming**

While still on the outer edge of marketing budgets (games' share in the overall advertising spending remained at a small 0.1%), in-game advertising and advergaming is expected to grow to a billion dollar business by 2009 [29]. Advergaming can be described as the practice of using games to advertise or promote a product, organization or viewpoint. Both in-game advertising and advergaming are a natural and increasingly accepted evolution from print ads, T.V. ads, and banners, taking full advantage of the interactivity of the Internet and involving consumers in ways not possible with other media.

- **Growing virtual economy**

The virtual marketplace is a very popular Internet application for a number of years now. While tangible products are still by far the most traded items on online auction sites such as eBay, virtual products and services are becoming increasingly popular. It is today considered common practice that goods and services within virtual economies (mainly MMORPGs) are sold on online auction sites and are traded for real currencies. Since players of these games are willing to substitute real economic resources of time and money (monthly fees) in exchange for these resources, by definition they have demonstrated utility to the user. In fact the first cyber real estate agents are already running a profitable business in the virtual world of 'Second Life' [35].

4 Part I: Gaming Performance Model

As explained earlier, the gaming performance queuing model is an extension to the model described in the paper [5] “Predicting Ping times in FPS games”. The original paper follows a theoretical approach for assessing the delay performance that could be attained under certain traffic assumptions. In this chapter we will use this same approach to predict ping times in FPS games, but in addition the impact of VoIP traffic will be included. Before starting with the mathematical modeling, let us give an overview of the problem background.

4.1 Introduction

Since the birth of the Internet years ago, a variety of services have been deployed over this globally accessible network. Originally, the majority of those consisted of data-driven services, such as e-mail and file-sharing applications. However with the ongoing penetration of broadband Internet access real-time services like VoIP, video streaming and online gaming have become increasingly popular. Unfortunately there is no QoS guarantee in the current best effort Internet model; traffic stemming from interactive services is likely to mix with other traffic running over the Internet. In the worst case it interferes with data streams of TCP sources, which tend to increase their sending rate until a part of the network is congested. For that reason The Internet Engineering Task Force (IETF) has defined two different approaches to support Quality of Service in IP networks:

- **Integrated Services (IntServ)**
The idea of IntServ is that every router in the system implements IntServ, and every application that requires some kind of guarantees has to make an individual reservation. The problem with IntServ is that many states must be stored in each router. As a result, IntServ works on a small scale, but as it scales up to a system the size of the Internet, it is difficult to keep track of all of the reservations.
- **Differentiated Services (DiffServ)**
Instead of classification per-flow, DiffServ classifies packets into classes, depending on the value of the codepoint in the packet’s IP header. DiffServ does not provide absolute guarantees, but instead each router acts on classes depending on a Per-Hop Behavior.

But even when interactive and elastic services are segregated, different interactive services (each with their own delay requirements) may interfere in nodes of the network, leading to a possible violation of those requirements. The impact of running multiple real-time traffic streams over the same network will be the main research subject of this chapter.

A number of different processes contribute to the end-to-end delay of an online gamer. These delay contributions can be divided into three different groups [9]:

- Delay caused by hardware and software that registers and processes user input
- Network delay introduced by sending gaming data (including user input) over the Internet
- Rendering delay introduced by hardware and software responsible for updating the virtual world (both client- and server side)

A recent study [12] indicates that most users attribute the large part of their gaming experience disruptions to network related problems. For that reason and the fact that the other two delays can be minimized by using PC configurations with last generation hardware, only the network delay will be discussed in this report. The network delay itself can be split into several components [9]:

- Propagation delay
- Packet processing delay
- Serialization delay
- Queuing delay

Of these components the first three delays are deterministic and relatively easy to assess. The queuing delay is harder to assess because of its stochastic nature. Studies in the past have shown that deterministic upper bounds can be obtained, but this worst-case approach leads to unrealistically high delay values. Therefore delay quantiles will be derived, leading to more realistic values [5].

The choice of FPS games out of the large pool of multiplayer games is all but random. Various recent studies (e.g. [6]) have shown that within this genre of multiplayer games the quality experience of users is extremely sensitive to delay. In order to respect the strict delay requirements of these services, the generated data is transported over the UDP protocol. UDP does not adjust the speed of packets being sent onto the network, even when the network is congested. TCP will re-send any dropped or corrupted packets, introducing jitter or variation in delay. This will reduce the quality of the game. Although the player would have perfect information, they would find that the game would pause whenever packets were lost as TCP tries to recover from the error.

4.2 Traffic Source model

The original Gaming Performance model is based on a traffic source model which describes the main traffic characteristics of modern FPS games. The traffic flows of a typical FPS are generated in the following way [5]: a central dedicated server keeps track of the global gaming state. At timer-based intervals it sends a burst of packets containing this gaming state to the clients. The clients then receive and process these packets to update the view on the screen. Afterwards the client's input is processed and an update packet is sent back over the network.

The traffic source model itself is based on a combination of papers published on the subject [3, 7] as well as research that was carried out by the authors of [5]. Earlier conducted traffic traces, discussed in the related work section of [5], have been approximated by a variety of distribution functions, including the shifted Weibull, the lognormal and the extreme distribution. It becomes clear that even within the FPS genre traffic characteristics can show large fluctuations. The traffic source model can be divided into two parts: the client traffic model (upstream direction) and the server traffic model (downstream direction). The characteristics of the traffic traces conducted by the authors of [5] are summarized below:

Client -> Server traffic			
	<i>Mean</i>	<i>CoV</i>	<i>Distribution</i>
Inter Arrival Time [ms]	30	0.65	Deterministic
Packet Size [byte]	73	0.06	Deterministic

Server -> Client traffic			
	<i>Mean</i>	<i>CoV</i>	<i>Distribution</i>
(Burst) Inter Arrival Time [ms]	47	0.07	Deterministic
Packet Size [byte]	154	0.28	-
Burst Size [byte]	1852	0.19	Erlang(K)

The above tables also include four distribution fits. These distributions are chosen by the authors of the original Gaming Performance Model and are based on both their own measurements as well as earlier fitted traffic data of various FPS games. The source model constructed in [5] is unique in the sense that the distribution functions of the server traffic model are fitted on tail behavior instead of the first two central moments. However, the limited size of their traffic measurements has to be taken into account.

The authors have chosen to model the size of the bursts sent by the server with an Erlang distribution, with general K . The number of gamers appears to only affect the mean burst size; the CoV of the burst size is insensitive to the number of gamers. The choice of the Erlang distribution is not only a consequence of the tail fit but it also enables the authors to find a closed form expression for the waiting time distribution.

4.3 Upstream

The above traffic source model contains all the information needed to start with the construction of our mathematical model. Following the structure of this underlying traffic model, we have divided the model in an upstream and downstream component. The assumptions on inter-arrival and service times will be the building blocks of our mathematical queuing model. Later in this chapter both separate parts will be merged again to arrive at the desired queuing model, which will enable us to calculate ping times. For now let us start with the upstream direction.

We are to analyze the situation in which two Constant Bit Rate streams interfere (Figure 4). The upstream queuing delay is the result of the competition of packets stemming from VoIP users and gamers on the link to the server. It may happen that a packet from a certain client arrives on that aggregation point at the instant a packet from another client is being transported over the link towards the server. As a result, a queue can temporarily build up in that part of the network. We assume that on the long run fewer packets arrive than can be served (per time unit) and therefore this queue should regularly empty.

It is shown that we are really looking at the $M/G/1$ queue when the number of users grows large [4, 5]. So in order to calculate the upstream queuing delay it is possible to model the aggregation node as an $M/D_1+D_2/1$ queuing system instead of using the more complex $N_1*D_1+N_2*D_2/D_1+D_2/1$ system. Modeling this aggregate as a Poisson process enables us to derive an upper bound for ping times, since the real arrival process will likely show a lot less variation. Empirical observations [16] indicate that this worst case approximation is very accurate when we are examining situations with relatively low loads.

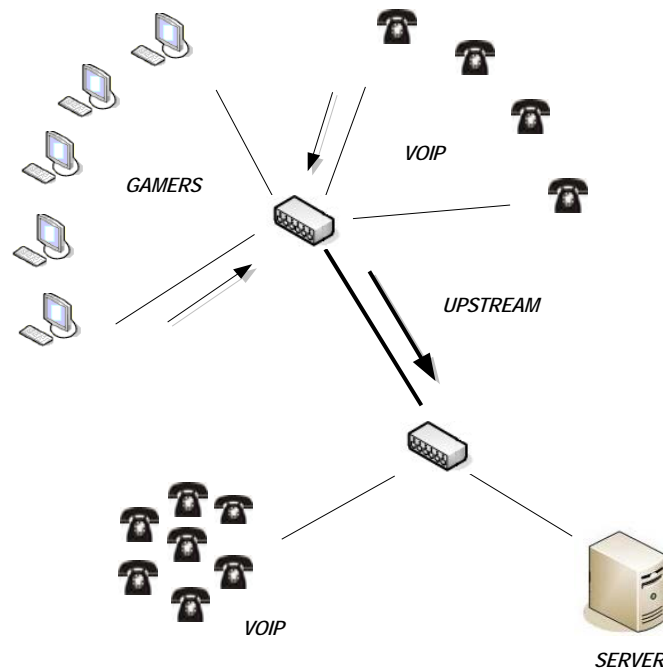


Figure 4: Upstream situation

As stated in the introduction of this chapter, we have chosen to derive delay quantiles instead of deterministic upper bounds. This approach enables us to calculate more realistic delay values. In order to calculate these delay quantiles, let us define the moment generating function (mgf) of a random variable d as

$$D(s) = Ee^{sd}. \quad (0)$$

In both [4] and [5] it is stated that the mgf of the delay in a M/G/1 queuing system can be approximated as follows

$$D_u(s) \approx 1 - R + R \frac{\lambda}{\lambda - s}. \quad (1)$$

We use this approximation instead of the exact mgf of the waiting time distribution (which can be found in e.g. [11]), because it will later enable us to derive a closed form expression for the mgf of the total delay. In equation (1), λ refers to the dominant pole of the exact mgf of the waiting time distribution. Its value follows from solving the following equation

$$\rho(B(\lambda) - 1) = \lambda B'(0), \quad (2)$$

where $B(\cdot)$ denotes the mgf of the service time and ρ stands for the load. The load in a M/G/1 queue is defined as the mean arrival intensity multiplied by the mean service time [19]. Once λ is determined we can determine the last unknown parameter R of equation (1) by evaluating equation (3)

$$R = \frac{(1 - \rho)}{\rho \frac{B'(\lambda)}{B'(0)} - 1}. \quad (3)$$

Equations (1), (2), and (3) provide the framework for the approximation of the mgf of the delay in the upstream direction. However, we first need to describe the traffic characteristics of our uplink M/G/1 queuing system. We introduce the following parameter notation (Table 1).

Table 1: Upstream M/G/1 parameter notation

	Gaming	VoIP
No. Users	N_1	N_2
Mean IAT [ms]	D_1	D_2
Mean Service time [ms]	S_1	S_2
Aggregation link rate [kbit/s]		R_u

Now let P_1 and P_2 denote the fraction of packets with service time S_1 and S_2 .

$$P_1 = \frac{N_1 / D_1}{N_1 / D_1 + N_2 / D_2}, \quad P_2 = \frac{N_2 / D_2}{N_1 / D_1 + N_2 / D_2}. \quad (4)$$

To simplify calculations a scaling of the actual service times is applied. S_1 and S_2 are expressed as multiples of the minimum service time in the entire system (including the service times in the downstream direction). The packet sizes of the two different packets are deterministic and thus because of the constant link speed the service times are deterministic as well. We therefore know that the mgf of the combined service time, its derivative and the derivative's value at $s=0$ are given by

$$B(s) = P_1 e^{-sS_1} + P_2 e^{-sS_2}, \quad (5)$$

$$B'(s) = P_1 S_1 e^{s \cdot S_1} + P_2 S_2 e^{s \cdot S_2}, \quad (6)$$

$$B'(0) = P_1 S_1 + P_2 S_2. \quad (7)$$

The uplink load is determined by the formula

$$\rho_u = \frac{8N_1 \frac{S_1}{D_1} + 8N_2 \frac{S_2}{D_2}}{R_u}. \quad (8)$$

We use the notation ρ_u for the uplink load (and ρ_d for the downlink load in the next section) instead of ρ . Equation (8) can be intuitively explained in the following way: the uplink load can be defined as the total bit rate generated by the voice and gaming traffic divided by the aggregation link rate. The sum in the numerator is multiplied by 8 because the link rate R_u in the denominator is expressed in bits per second, while the packet sizes S_1 and S_2 in the numerator are expressed in bytes.

We are now ready to determine the values of λ and R by filling in and solving the equations (2) and (3)

$$\rho_u (P_1 e^{\lambda \cdot S_1} + P_2 e^{\lambda \cdot S_2} - 1) = \lambda (P_1 S_1 + P_2 S_2). \quad (9)$$

$$R = \frac{(1 - \rho_u)}{\rho_u \frac{P_1 S_1 \cdot e^{\lambda \cdot S_1} + P_2 S_2 \cdot e^{\lambda \cdot S_2}}{P_1 S_1 + P_2 S_2} - 1}. \quad (10)$$

Once λ and R are determined, the mgf of the upstream delay is approximated by equation (1) as stated earlier

$$D_u(s) \approx 1 - R + R \frac{\lambda}{\lambda - s}. \quad (11)$$

4.4 Downstream

In this section we will derive an expression for the mgf of the downstream delay. Our mathematical model is again based on underlying traffic characteristics, which stem from the traffic source model of Section 4.2.

We observe that a tagged packet arriving in a batch of gaming packets experiences a delay $d_{n,t}$ composed of two contributions

$$d_{n,t} = w_n + p_{n,t}. \quad (12)$$

First, the tagged packet has to wait for all the remaining work in the system the instant it arrived, which is equal to the delay w_n the batch sees. Second, it has to wait a time $p_{n,t}$ due to packets that arrived in the same batch and that are in front of the tagged packet.

Let us define the moment generating functions $D(s)$ and $P(s)$ for the steady state random variables $d_{n,t}$ and $p_{n,t}$ respectively. Since w_n and $p_{n,t}$ are statistically independent, we get

$$D_d(s) = W(s)P(s). \quad (13)$$

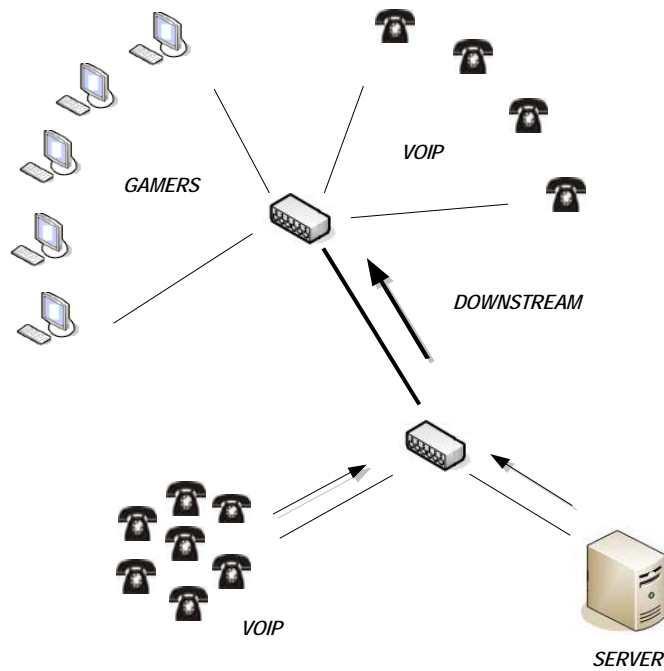


Figure 5: Downstream situation

4.4.1 Delay distribution of batches

The gaming server sends batches of gaming packets back over the network at a constant rate. This rate is server-side configurable in most modern First Person Shooters. Server administrators often pick values in the range of 25-50 milliseconds. Low inter-departure times result in a responsive, fluid experience for gamers, but at the cost of higher bandwidth usage. The batch sizes are modeled as i.i.d. random variables with an Erlang distribution of order K as marginal distribution. These batches interfere with small voice packets stemming from a certain number of VoIP users (Figure 5). The VoIP traffic is again modeled as a CBR stream and thus the resulting queuing can be described by the $D+M/E_k+D/1$ system, which falls in the category of $G/G/1$ queuing theory. Few explicit formulas are known for these systems; especially tail probabilities are hard to derive.

Simulation

In order to get the necessary insight into the behavior of the $D+M/E_k+D/1$ queue, we have implemented a simulation model in the Matlab environment. During simulation we have used the following set of traffic parameter values (Table 2; N_1 and N_2 are determined after we have chosen the load and the fraction VoIP).

Table 2: Traffic parameter values

	Gaming	VoIP
No. Users	N_1	N_2
Mean IAT [ms]	40	40
Mean Packet size [ms]	125	160
Aggregation link rate [kbit/s]		5000

Simulation of the G/G/1 system shows some very interesting behavior. When we add VoIP users and simultaneously lower the number of Gamers (such that the downstream load stays the same), the 0.999 quantile of the one-way delay increases instantly with only the slightest interference with VoIP traffic (Figure 6).

The global behavior of the system is similar for different values of the Erlang order K and the downlink load ρ_d . The magnitude of the observed ‘jump’ is however dependent on the two previously mentioned parameters; the jump is magnified for larger values of K , while an increasing load has a diminishing effect. The shape of the curve can be intuitively explained in the following way:

When the Fraction VoIP (Δ) is zero, we are really looking at the D/E_k/1 queue and the simulation results coincide with the analytical solution of this queue, as derived in [5].

When we add a little bit of VoIP traffic, Δ becomes non-zero and the large gaming bursts start to interfere with small voice packets with relatively high arrival intensities. The mean service time of a gaming burst is large compared to the mean inter-arrival times of the voice packets, so during the regular service time of a gaming burst, many voice packets are being queued in the buffer. On the other hand very few gaming bursts are being queued, due to the small deterministic service times of the voice packets. Once the next gaming burst arrives, all voice packets have received service and consequently the burst finds the buffer empty most of the time.

Following the x axis of the graph, the traffic in the system consists of more and more VoIP traffic. The phenomenon described above becomes less dominant, because the mean size of the gaming bursts decreases (and therefore service times as well because of the constant link speed). Although mean inter-arrival times of voice packets decrease, less of those packets have to wait on gaming bursts because of this decrease in burst service times.

Eventually Δ reaches 1. This is the scenario in which traffic only consists of VoIP traffic. Following the reasoning of for example [4], this situation corresponds to the M/D/1 queue. Delay quantiles of this queue are well known and are similar to our simulation results.

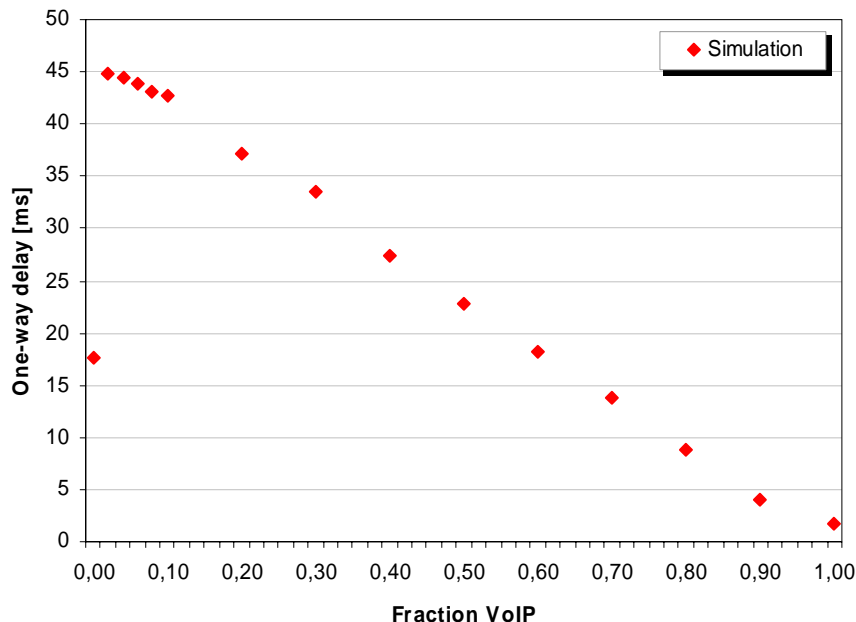


Figure 6: Simulation of G/G/1 delay quantiles

M/G/1 approximation

Now let us start our queuing analysis with the assumption that the aggregate arrival process can be approximated with a Poisson process. This assumption enables us to derive the delay distribution of batches making use of the $M/E_k+D/1$ system, instead of the more complex $D+M/E_k+D/1$ system. Modeling an aggregate as a Poisson process is again worst case and the accuracy of our assumption will become evident when we compare the results to our simulation.

Analog to the uplink queuing system, the downlink $M/G/1$ queuing system can be described using the parameter notation of Table 3.

 Table 3: Downlink $M/G/1$ parameter notation

	Gaming	VoIP
No. Users	N_1	N_2
Mean IAT [ms]	D_1	D_2
Mean Service time [ms]	S_1	S_2
Aggregation link rate		R_d

Let P_1 and P_2 again denote the fraction of packets with service time S_1 and S_2

$$P_1 = \frac{1/D_1}{1/D_1 + N_2/D_2}, \quad P_2 = \frac{N_2/D_2}{1/D_1 + N_2/D_2}. \quad (14)$$

P_1 and P_2 are calculated in almost the same way as the fractions in the uplink direction, but they do differ in one important way. The difference lies in the fact that all the gaming packets are sent back over the network in one burst. Therefore, if we want to calculate the probability that a randomly arriving packet belongs to a certain class (either gaming or voice), we have to use 1 instead of N_1 in both fraction formulas (14). S_1 and S_2 are again expressed as multiples of the minimum service time in the entire system (including the service times in the upstream direction).

From the traffic source model we know that the service times of the voice packets are deterministic while the service times of the gaming bursts can be described with an Erlang distribution. As a result the mgf of the combined service time, its derivative and the derivative's value at $s=0$ are given by

$$B(s) = P_1 \left(\frac{\beta}{\beta - s} \right)^K + P_2 e^{s \cdot S_2}, \quad (15)$$

$$B'(s) = P_1 \frac{K\beta^K}{(\beta - s)^{K+1}} + P_2 S_2 \cdot e^{s \cdot S_2}, \quad (16)$$

$$B'(0) = P_1 \frac{K}{\beta} + P_2 S_2. \quad (17)$$

In the above formulas β is defined as the Erlang order K divided by the mean service time of a gaming batch. Equations (2) and (3) again determine the unknown parameter values of our mgf approximation. However, for the downstream direction we introduce a slightly different notation. The downstream load is now denoted by ρ_d and is determined (in similar fashion to the upstream situation) by the following equation

$$\rho_d = \frac{8N_1 \frac{S_1}{D_1} + 8N_2 \frac{S_2}{D_2}}{R_d}. \quad (18)$$

The dominant pole λ in equation (2) is now called γ to avoid notation problems when we arrive at the derivation of the total delay. Its value follows from equation (19),

$$\rho_d \left(P_1 \left(\frac{\beta}{\beta - \gamma} \right)^K + P_2 e^{\gamma \cdot S_2} - 1 \right) = \gamma \left(P_1 \frac{K}{\beta} + P_2 S_2 \right). \quad (19)$$

Instead of R in the upstream direction, we now use C in the downstream direction to refer to the last unknown parameter of our mgf approximation,

$$C = \frac{(1 - \rho_d)}{\rho_d \frac{P_1 \frac{K\beta^K}{(\beta - \gamma)^{K+1}} + P_2 S_2 \cdot e^{\gamma \cdot S_2}}{P_1 \frac{K}{\beta} + P_2 S_2} - 1}. \quad (20)$$

Once γ and C are determined, the mgf of the downstream burst delay $W(s)$ is approximated in the following way:

$$W(s) \approx 1 - C + C \frac{\gamma}{\gamma - s}. \quad (21)$$

Let us compare the simulation of the original D+M/Ek+D/1 system with the M/G/1 approximation.

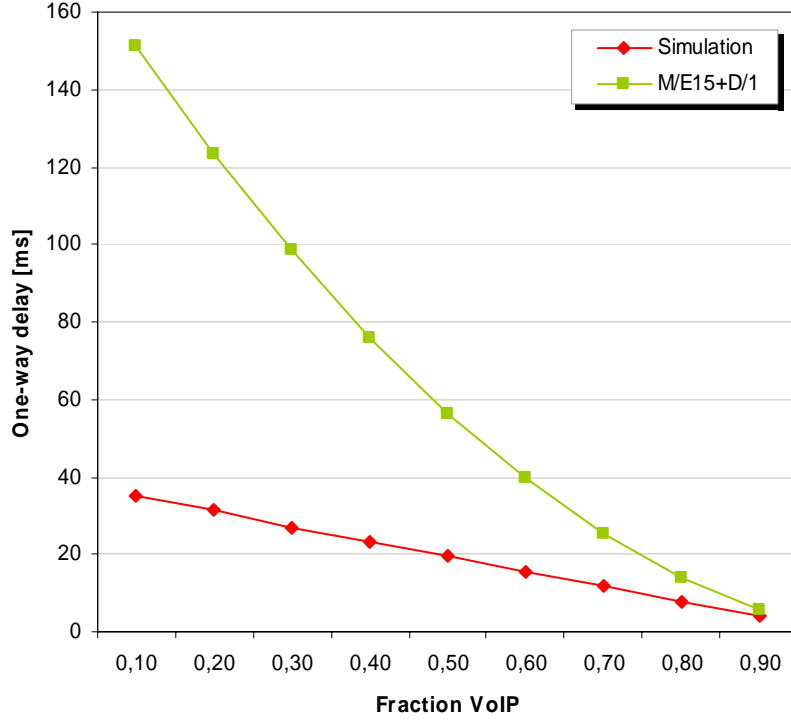


Figure 7: Comparison of M/E15+D/1 and simulated G/G/1 quantiles (K = 15, Load = 0.6)

Because we are really interested in the situation in which traffic consists of both voice and gaming packets, we will only consider fractions VoIP between 0.1 and 0.9. The traffic parameters are again determined by Table 2. Figure 7 clearly demonstrates that our approximation leads to unrealistically high delay quantiles. Therefore we have come up with a way to scale our M/G/1 results.

We have chosen to divide the M/G/1 delay quantiles by our simulated quantiles. Subsequently regression analysis was carried out on those quotients, based on the factors fraction VoIP, downlink load, and the Erlang order. All of our simulation results are based on the combinations of the following factor values (in combination with the traffic values summarized in Table 2):

- Fraction VoIP Δ : 0.1 – 0.9 (steps of 0.1)
- Downlink load ρ_d : 0.4, 0.6, 0.8
- Erlang order K : 2, 9, 15

Ultimately we have chosen to use the built-in Excel LINEST function, which is based on the Least Squares Method to calculate the line that best fits the quotients. Based on regression statistics provided by the same LINEST function we have chosen not to include an intercept, as the coefficient of determination R^2 (ratio of the explained variation to the total variation) of that model was considerably lower.

$$S = \frac{M/G/1}{Sim} = 5.792\rho_d + 0.115K - 3.172\Delta . \quad (22)$$

The above linear regression model has a coefficient of determination of 0.932, which means that 93.2% of the total variation in the quotients can be explained by regression equation (22). The t-values of the factors ρ_d , K , and Δ are, respectively, 17.841, 7.316, and -11.230. The critical t-value, two tailed, with 87 degrees of freedom and Alpha = 0.05 is 2.447. We can therefore conclude that all three independent factors are statistically significant.

Now let us take a look at two test scenarios:

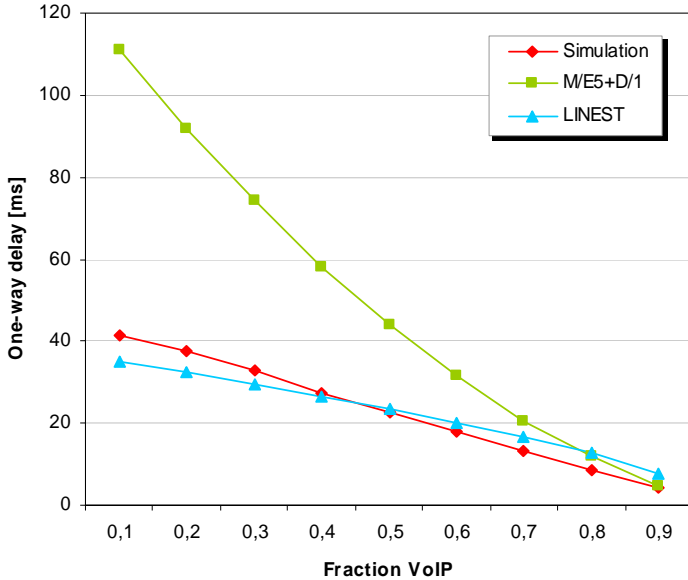


Figure 8: Test scenario 1 ($K = 5$, Load = 0.5)

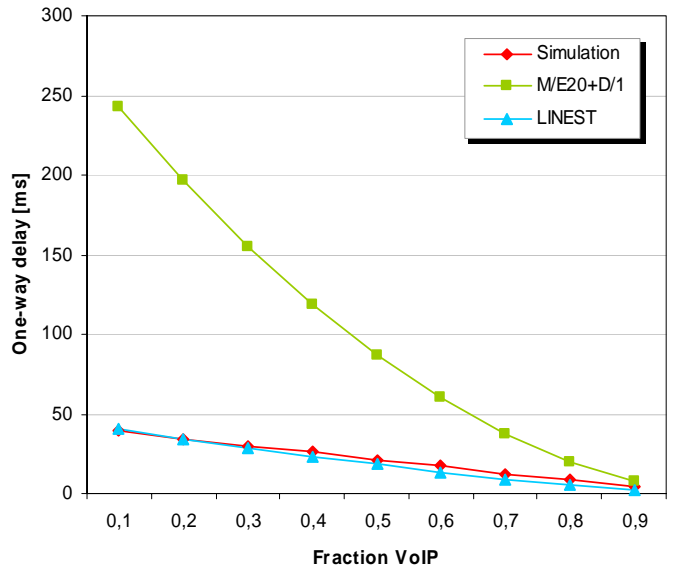


Figure 9: Test scenario 2 ($K = 20$, Load = 0.7)

When we use our regression model to scale M/G/1 quantiles of new scenarios and subsequently compare them to new simulated quantiles (which have not been used for fitting of the regression line),

results look good (Figure 8 and 9). Especially scenarios with equal amounts of VoIP and gaming traffic are remarkably well approximated, with absolute errors in the range of 0-3 milliseconds.

In the remainder of the paper we will use the scaled version of the M/G/1 moment generating function, in which the original dominant pole γ is multiplied by the scaling factor S and C is still determined by equation (20) based on the original dominant pole

$$W(s) \approx 1 - C + C \frac{\gamma}{\gamma - s}. \quad (23)$$

Because of the complexity of the G/G/1 queue we had to resort to the above approximation. During simulation it became clear that the downstream delay quantiles are heavily dominated by the VoIP component. Another logical approximation approach would be to model the downstream situation as a queue with server vacations. So instead of considering both types of traffic, the service periods of gaming bursts could be modeled as off periods of the aggregation link. Because of time restrictions this exploration of this approach is left for future research. The thesis [14] of R.N. Queija contains an extensive part on queuing with server vacations. Although he discusses sojourn times of processor-sharing queues instead of waiting times of First Come First Serve queues, his results may be useful for future work.

4.4.2 Delay distribution of packets

Let us recall that $P(s)$ is defined as the moment generating function of the delay distribution of a certain tagged packet. The delay is caused by other packets that arrive in the same batch and that are in front of the tagged packet. To assess $P(s)$ we need to know where in the arriving batch the tagged packet resides. This is described by a random variable $u_{n,i}$ with pdf $p_u(u_{n,i})$ with support $[0,1]$. $u_{n,i}=0$ means that the tagged packet is the first of the n -th batch, while $u_{n,i}=1$ means that the packet is the last one of the batch and has to wait for (practically) the entire batch. We assume that there are as many packets in a batch, such that the size of a packet is negligible with respect to the size of the batch, and hence $u_{n,i}$ can take any value in $[0,1]$. By first conditioning on the batch size, then averaging over all possible batch sizes and a change of variables, it follows that

$$P(s) = \int_0^1 \left(\frac{\beta}{\beta - s\tau} \right)^K p_u(\tau) d\tau. \quad (24)$$

For general distributions $p_u(u_{n,i})$, no conclusions can be drawn with respect to the poles of $P(s)$, but we consider the following special case.

Let us assume the packet can reside anywhere in the batch, i.e., $p_u(u_{n,i})$ is the uniform distribution. In this case, the integral defining $P(s)$ can be calculated (by identifying a primitive function of the integrand)

$$P(s) = \frac{\beta}{(K-1)s} \left[\left(\frac{\beta}{\beta - s} \right)^{K-1} - 1 \right] \quad K > 1. \quad (25)$$

This can be written as a weighted sum of Erlang terms (using Horner's rule) which leaves us with the following mgf [5]

$$P(s) = \frac{1}{(K-1)} \sum_{k=0}^{K-2} \left(\frac{\beta}{\beta - s} \right)^{K-1-k} \quad K > 1. \quad (26)$$

4.5 Total delay

In the previous paragraphs the moment generating functions of both the up- and downstream delays were derived. In order to calculate the desired ping times the mgf of the total delay is still to be determined. Because we assume the up- and downstream delay to be independent, this mgf $T(s)$ is equal to the product $D_u(s)D_d(s)$, which is equal to $D_u(s)W(s)P(s)$. We thus have

$$T(s) = D_u(s)W(s)P(s). \quad (27)$$

Making use of the technique described in Appendix A of [5] it can be shown that each term in $T(s)$ can be written as a sum of Erlang terms. Therefore it is also possible to write the product as a sum of Erlang terms. This leads to

$$\begin{aligned} T(s) &= (W(\lambda)P(\lambda)R) \left(\frac{\lambda}{\lambda-s} \right) + (D_u(\gamma)P(\gamma)C) \left(\frac{\gamma}{\gamma-s} \right) \\ &+ \frac{1}{(K-1)} \sum_{k=0}^{K-2} \left(\sum_{l=0}^k (-\beta)^l \sum_{m=0}^l \frac{D_u^{(m)}(\beta)W^{(l-m)}(\beta)}{m!(l-m)!} \right) \left(\frac{\beta}{\beta-s} \right)^{K-1-k}. \end{aligned} \quad (28)$$

Inverting $T(s)$ gives us the Tail Distribution Function of the total delay, with the TDF defined as in [4]:

$$T_{delay}(x) = P[d > x], \quad (29)$$

i.e., the $(1-Q)$ -quantile is the (smallest) value x for which $T_{delay}(x) \leq Q$.

So in order to calculate delay quantiles this TDF needs to be inverted. Because of the scaling of the service times earlier on, this quantile value still has to be multiplied by the minimum of service times in the system. The ultimate queuing model implementation is partly shown in Figure 10. The entire Mathcad implementation can be found in Appendix A.

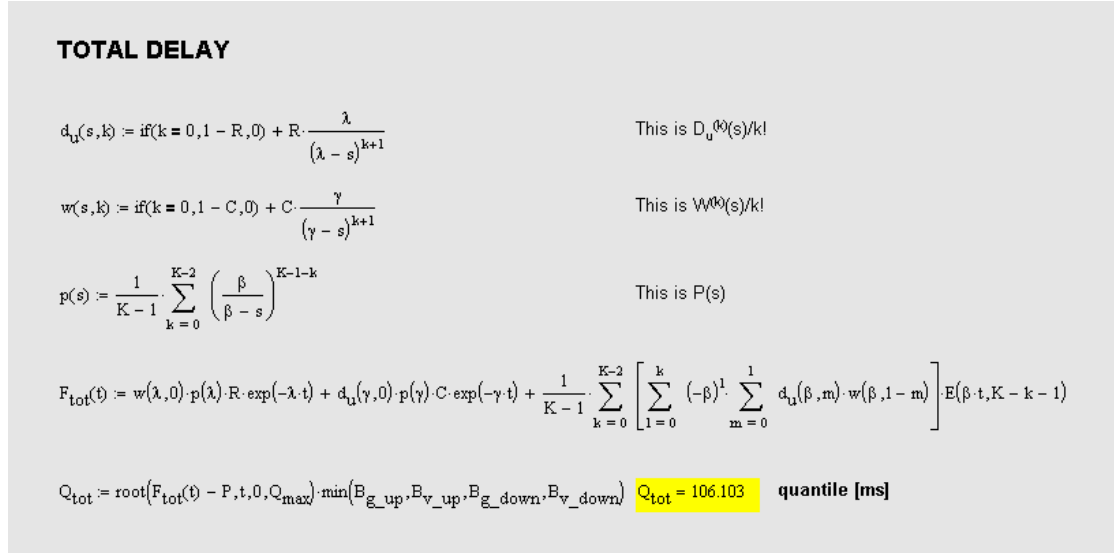


Figure 10: Mathcad calculation sheet

4.6 Results

In this section the Mathcad implementation of the total delay formula (shown in Figure 10) is used to calculate delay quantiles for a number of different scenarios. The scenarios are chosen such that the influence of different model parameters on estimated ping times will become evident.

In particular the influence of the following parameters will be assessed:

- Fraction VoIP
- Load on the aggregation link
- Inter-arrival time of gaming bursts
- Erlang order K of mean burst sizes

Our initial goal was to develop a generic model, which would enable us to calculate ping times for every possible combination of both network and traffic parameters. However during the analysis of the downstream delay of bursts (Section 4.4.1) we encountered some problems and had to resort to the stated approximation. The consequence of this approach is the decrease of generality of our Gaming Performance Model, because the scaling factor S is fitted on a number of pre-determined scenarios. Of course we have chosen parameter values which resemble real world values (known from related studies) as close as possible. The same parameter values are used in the scenarios below and are summarized in the following tables:

Table 4: Upstream parameter values

	Gaming	VoIP
No. Users	N_1	N_2
Mean IAT [ms]	40	40
Mean Packet size [byte]	80	160
Access link rate [kbit/s]		256
Aggregation link rate [kbit/s]		5000

Table 5: Downstream parameter values

	Gaming	VoIP
No. Users	N_1	N_2
Mean IAT [ms]	40	40
Mean Packet size [byte]	125	160
Access link rate [kbit/s]		1024
Aggregation link rate [kbit/s]		5000

All of the ping times are based on the 99.9% quantile of the queuing delay, because this quantile was also used during the simulation of the G/G/1 downlink queue. In order to resemble the real world situation as close as possible, all of our results include a serialization delay, based on the chosen link rates and packet sizes (see Tables 4 and 5 above).

Fraction VoIP

Figure 11 shows the influence of the fraction of VoIP on ping quantiles for different values of the Erlang order K . The number of gamers and VoIP users are chosen such that the downlink load remains constant at 0.5.

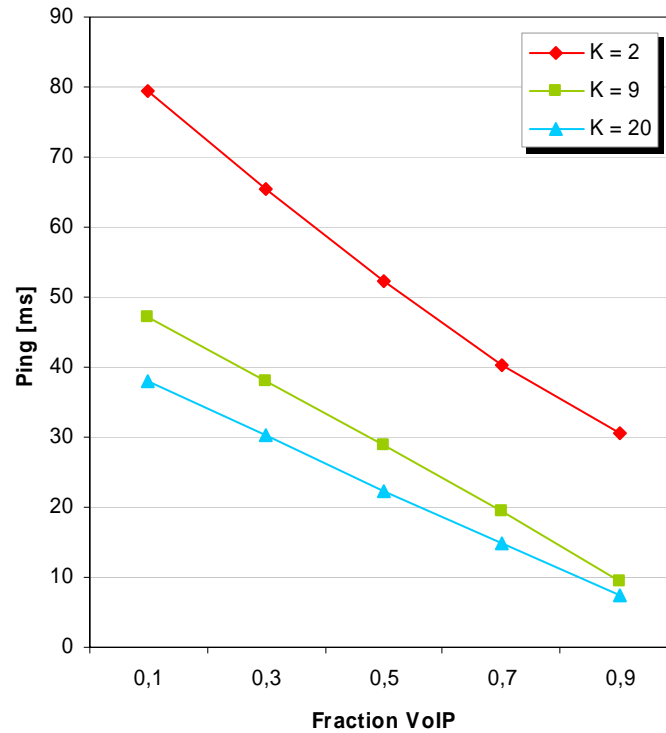


Figure 11: Influence of fraction VoIP on ping

It becomes clear from Figure 11 that there is a negative approximately linear relation between the fraction of VoIP and ping quantiles for all three considered Erlang orders K . When we take a closer look at the different delay components in the Mathcad calculation sheet we find that the downstream delay caused by queuing of gaming batches and voice packets is most dominant. As a result the above graphs show the same decreasing behavior as was shown for the one-way downstream delay of bursts in the figures of Section 4.4.1.

Load on the aggregation link

Figure 12 illustrates the influence of the load on the aggregation link on ping quantiles for different values of the Erlang order K . The number of gamers and VoIP users is chosen such that the downlink load consists of an equal amount of voice and gaming traffic (Δ is 0.5).

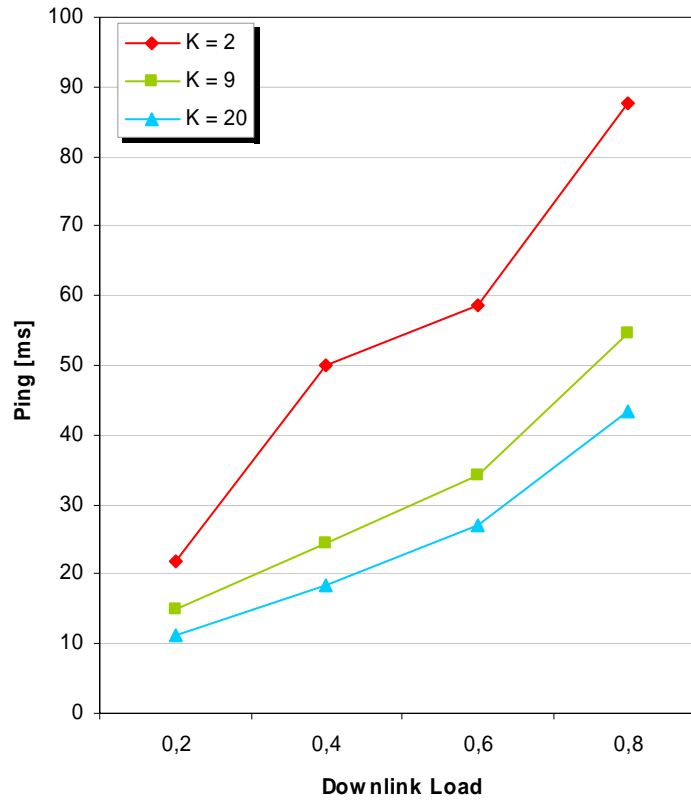


Figure 12: Influence of downlink load on ping

The graphs in Figure 12 confirm the expected positive relation between the downlink load and ping times. Remarkable is the behavior of the graph of $K = 2$. The observed twist for small downlink loads in combination with a small Erlang order is caused by our scaling factor S (Formula 22). For such scenarios its value becomes smaller than 1 and as a consequence $M/G/1$ delay quantiles are augmented, which is clearly unwanted behavior of our model. In order to deal with this problem an IF-statement is introduced in the Mathcad sheet such that S cannot drop below a minimum value of 1.

Inter-arrival time of gaming bursts

In Section 4.4.1 we already mentioned the option of game server administrators to set the frequency of server updates. The impact of altering this setting is shown in Figure 13.

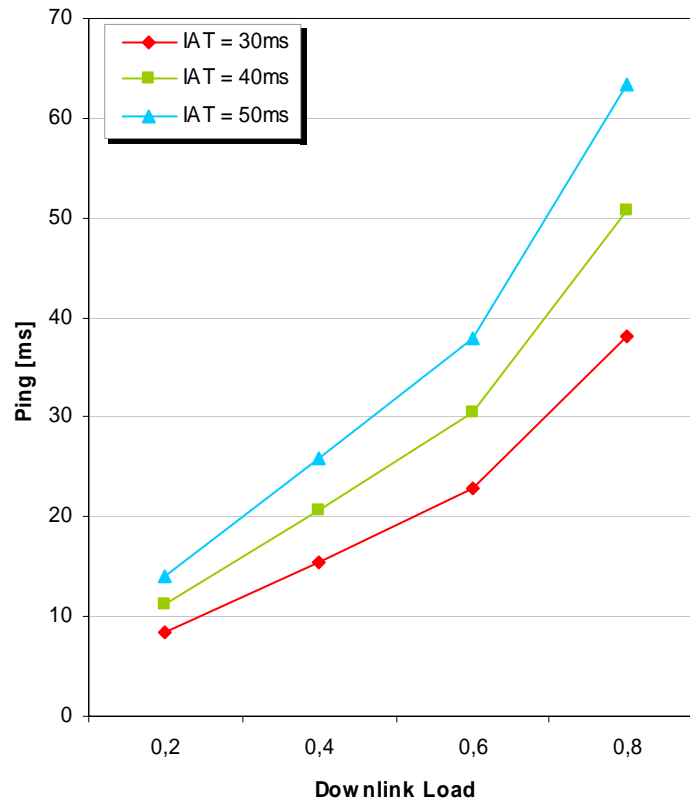


Figure 13: Influence of burst IAT on ping ($K = 9$)

We conclude that an increase of the burst IAT leads to higher ping quantiles. We can even conclude that the ping time is proportional to the IAT if the downlink delay contribution is dominant (when the downlink load is higher than the uplink load). For example the ping time of IAT = 50 ms is 5/3 times as high as the ping time of IAT = 30 ms. Similar results were reported in the results section of [4].

Erlang order K of mean burst sizes

The impact of the Erlang order K becomes apparent when we take a look at Figures 11 and 12. Evidently, the higher the Erlang order the lower the ping times. As expected the decrease in variation of the mean burst size results in less congestion in the system.

Now we have assessed the impact of the different model parameters, let us take a closer look at the impact of VoIP traffic and compare our results to the results of [5].

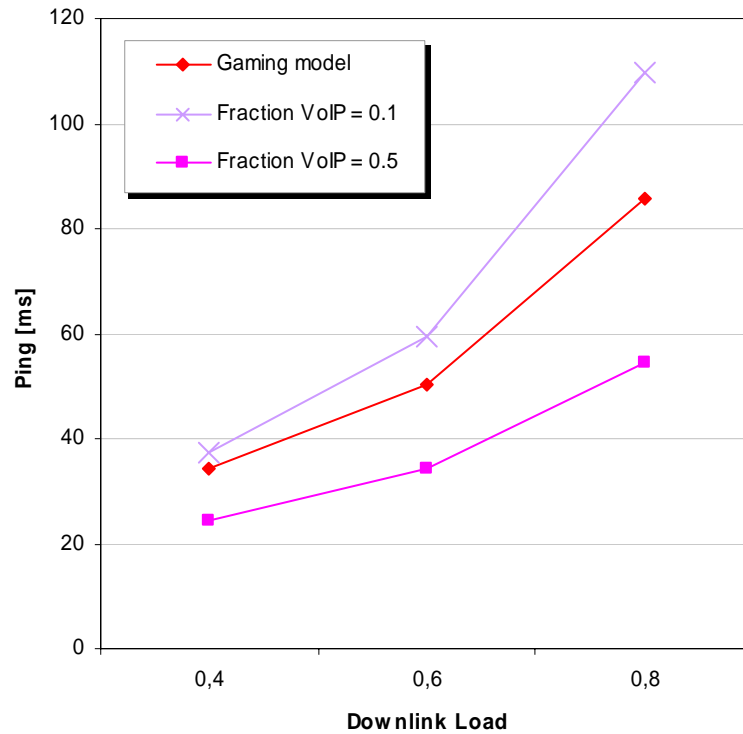


Figure 14: Pure gaming traffic vs. two mixed scenarios ($K = 9$)

It is interesting to see how the interference of VoIP traffic can lead to both a decrease as well as an increase of ping times, compared to the situation in which only gaming traffic runs over a network (middle line in Figure 14). Fractions of interfering traffic in the range of 10 - 25% of downlink bandwidth usage result in an increase in ping times again compared to results of the original model. The addition of larger fractions of voice packets leads to a relative decrease in ping times.

The above results can be used for dimensioning purposes of multi-service networks. For example if we take one of the scenarios plotted in Figure 14 (fraction VoIP = 0.5, IAT = 40ms, $K = 9$), we can derive that in order to realize a ping under 50 ms we can load the aggregation link to at most 75%.

5 Part II: G-model

In the first part of this report we have focused on the calculation of ping times of online FPS gamers. In this second part of our research project we will change our approach. Instead we will focus on capturing the gaming experience, based on a number of self-selected factors.

5.1 Introduction

Research on online gaming is relatively young. The last couple of years an increasing number of papers on the subject has been published, with large annual conferences like NetGames being a major boost for the gaming research community. From our own experience we know how important network circumstances are for an online gamer. Especially when gamers compete in online leagues and tournaments, a good connection to the server can give the little edge that makes the difference between winning and losing. It is therefore very important that both game hosting companies and Internet Service Providers understand the relationship between network conditions and the quality of the online gaming product. ISPs could use this valuable information to support adequate network provisioning and thereby optimally carrying this billion-dollar market segment. Moreover, insight into this relation enables independent server hosts and game developers to improve their specific marketing focus and customer support.

5.2 Related work

In order to get a clearer picture of the problem domain we have conducted a literature study on the quality experience of online gamers. The most important results of recent papers are summarized below.

In [12] the results of an online survey are presented. The objective of the survey is to provide insight to what users really think of the Internet and its impact on their playing experience. The approach differs from other conducted surveys, which have merely focused on characterizing the gaming population and understanding their motivation for playing. The authors have adopted a qualitative approach in the design of the survey instead of assessing the quantitative perception of users. The results of the survey indicate that most users attribute the majority of the disruptions in their gaming experience to network related problems. The large part of survey respondents is however not willing to pay for network QoS directly, because it is not a tangible product. Therefore the authors suggest an alternative approach, which integrates network compensation techniques in the game engine to decrease network related problems. Techniques like dead reckoning, client prediction and buffering are already present in most modern online games, but according to both the authors and the questioned gamers there is still a lot of room for improvement.

In the paper [10] the following hypotheses were tested: 1) higher levels of delay discourage players to join a server, because of the ping indicator in their server browser, 2) once connected to a server, the in-game indication of high delays encourages players to leave the server. The authors have monitored two public Half-Life servers for about one week while introducing additional delay. They have chosen to add delay in the range of 0-250 ms to degrade the network conditions. Although the experiments described in this paper are relatively short, some interesting results were found. The network delay appears to have a significant impact on a user's decision to join a game server, while players were not convinced to leave the server when confronted with additional delay. This last result is remarkable since players were able to notice the delay and their performance (kills per minute) was significantly worse.

In [2] the authors have focused on analyzing the impact of both delay and packet loss on user performance in FPS games. First typical real world values of packet loss and latency are deduced by monitoring public UT2003 servers. About 80% of all UT2003 servers have pings of less than 140. The maximum reported loss is just over 3%, with 80% of the servers showing no measurable loss at all. These values were used as guidelines for a controlled network experiment. The results of this experiment show that packet loss has no measurable effect on a player's performance, in contrast to latency which significantly decreases the shooting accuracy and number of kills for only modest amounts (75-100 ms). Subjective comments collected during the user studies show that latencies as low as 75 ms can result in a sluggish feel, while packet loss rates up to 3% are hardly noticed.

A similar network experiment is described in [13]. In this paper the authors analyze the objective and subjective influence of delay and jitter on the quality of gameplay of UT2003. The conclusions are consistent with earlier research on UT2003 and state that network impairment has a negative influence on both the player's perceived game quality and performance. The perceived quality depends on the size of the introduced delay, with ping times above 60 ms resulting in decreased gaming satisfaction. Jitter however does not play a prominent role in whether a player feels hampered.

The paper [20] proposes a technique for estimating the effects of network impairments on online game play. The authors stress that of all common home applications online gaming is the only one without an effective benchmark. Therefore they have come up with the Quake 3 G-model which calculates the impairment factor R , based on the factors latency, jitter and packet loss. R is then converted to the metric OPScore, using a lookup table which relates R to the number of kills (often denoted as frags) per minute. The metric OPScore is used because the authors argue that the number of frags per minute is not an appropriate figure of merit for all online games.

A very recent analysis of impairment factors can be found in [6]. In this paper the authors relate the influence of ping, jitter and skill level to players' perception and performance in four different multiplayer games (Counter Strike, Unreal Tournament 2004, Need for Speed, Warcraft 3). Furthermore, the subjective mean opinion score is introduced as well as the objective game outcome score as a normalized metric for multiplayer games. A multi-dimensional linear regression analysis shows that the various multiplayer games behave very differently under the same network conditions. This result is nowadays regarded as common knowledge, but this statement appears to be also true within the FPS category. This is possibly the result of different algorithms or implementations of prediction and latency compensation techniques present in today's games.

5.3 Factors that influence the perceived quality of FPS gamers

In the last section we have given an overview of existing literature on the quality experience of online gamers. Based on this literature study we have selected three impairment factors for further examination, namely ping, jitter and packet loss. In this section the choice for this set of factors will be motivated along with the factor values which will be examined in the user experiment (Section 5.4).

- **Ping**

Values: 0, 20, 40, 80, 160, 320 ms

This impairment factor needs no introduction. If we review results from former user studies, the ping time appears to be the single most important degrading factor of a gamer's quality experience, with latencies as low as 75 ms resulting in a sluggish feel. Some researchers even label ping times as culprit, because of built-in ping features of modern multiplayer games, with the ping value itself influencing the player's performance and joy.

In contrast to the authors of [20], who only consider the upstream delay, we have implemented round trip delay by dividing it evenly over both traffic directions. We have run some short tests in

our lab to test the impact of different distributions over the two directions, but no real differences were noticeable.

- **Jitter**

Values: 0, 10, 20, 40, 80, 160 ms (2 variants)

Jitter can be defined as the variance of the ping or delay. There are a couple of underlying causes which can result in a variation of ping times. Congestion of the home network caused by background traffic is one of these sources. Typical home applications like FTP transfers, video streams and browsing share a bursty nature. These bursts of traffic will sometimes place certain amounts of data between gaming packets, thereby leading to jitter in the game data stream. A second cause of jitter is variation in the traversed path of gaming packets, sometimes denoted as route flapping. Packets do not necessarily travel along the same nodes of the Internet and even the number of hops may vary from packet to packet. Because of the stringent delay requirements of FPS games, the implementation of jitter buffers is unfeasible. Although the authors of [13] conclude that jitter does not play a prominent role in whether a player feels hampered, research on this impairment factor is limited.

Inspired by the different causes of jitter, we have chosen to add variable delay to the network in two different ways. The first variant is implemented with the Pareto option enabled in a Netem script (the Netem network emulator will be described in Section 5.4.1). The Pareto distribution is used to approximate a number of different real world situations like the file size distribution of Internet traffic [30]. The second variant is implemented with a self-made script which resembles the occurrence of route flapping (see Figure 15). During intervals of 10 seconds the ping jumps between a base delay (determined by the current ping factor value) and a constant increased delay (determined by the jitter factor value). The impact of switching between different constant ping times is huge: while switching to the higher delay value (see arrow in Figure 15) the player movement is severely hampered. This degrading behavior also resembles the hampering a player experiences when someone starts a download session which fills the downstream link capacity (observed during tests in our gaming lab).

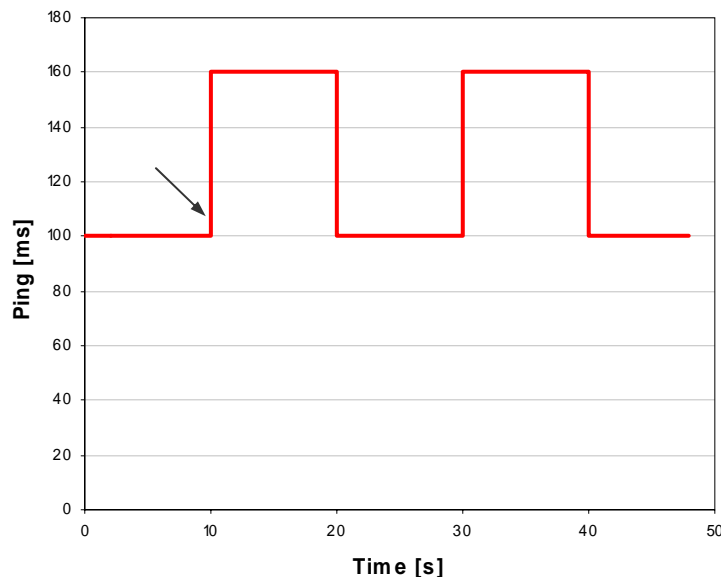


Figure 15: Route flapping scenario

- **Packet Loss**

Values: 0, 2, 5, 10, 20, 40%

There are arguments for both inclusion and exclusion of packet loss in our experiment. On the one hand, the majority of gaming servers show only very small amounts of loss, if they do at all. Furthermore according to the results of [2] packet loss goes unnoticed for typical real world values and has no measurable effect on a player's performance. However, if background traffic in the client's access network increases, troubles arise. When either the modem or client router becomes filled, gaming packets destined for the server are likely to be discarded. This is a realistic situation, since most home networks are nowadays used for a number of different services and by multiple users. Another trend is the wide adoption of wireless channels, which are more prone to packet loss than the traditional wire-line network environment. Ultimately, the popularity of the multi-service home network in combination with the growth of mobile gaming was decisive to take the factor packet loss into account.

Inspired by the upstream filling of modems and routers we have chosen to add packet loss only in the uplink direction, instead of dividing it evenly over both traffic directions. Although some quick tests with other distributions did not show noticeable differences in user experience, we felt that this implementation resembled real world scenarios as close as possible. Another argument is the fact that an equal distribution over the two directions would lead to a more complicated realization of mean factor values. For example, if we would add 10 percent packet loss in both the upstream and downstream direction to realize a total average packet loss of 20 percent, the effective average packet loss would be 19 percent ($0.10+0.10-0.10*0.10$) instead of the desired 20 percent.

Despite the fact that the number of frames per second at the client side has a huge impact on the gaming experience and is widely accepted as the number one indicator of hardware performance (in a gaming context), we will not include it in our experiment. By adding another additional factor, the size of the experiment would blow up. For now, we instead start with a selected number of network factors and focus on providing a robust framework for conducting gaming experiments which are reproducible. To our knowledge the combination of our three impairment factors have not yet been related to the subjective quality experience of gamers.

5.4 The experiment

The main objective of the experiment is to determine a relation between the perceived quality of gamers (expressed in Mean Opinion Scores) and the network related factors ping, jitter and packet loss. Because we require the experiment to be reproducible, we have used the guidelines of the paper [17] which provides a framework for performing subjective assessment of game quality.

We have chosen to use Quake IV [23, 25] for this user experiment, because it is one of the newest First Person Shooters on the market and has (by our knowledge) not yet been used for research purposes. As mentioned earlier, this multiplayer genre has very stringent delay requirements. Quake IV is very hardware demanding and generates a relatively large amount of traffic in the upstream direction, in comparison to most other FPS games. Although the game does not attract the same number of players as for example Counter Strike or Battlefield 2, it is currently used in many online gaming leagues and is therefore an interesting research subject.

Based on our own online gaming experience, we have chosen to discriminate between two different user types. We expect that players who primarily play for fun or are new to gaming, are less concerned with the quality of their connection than more serious players who for example compete in online competitions or are part of organized gaming clans. This expert factor is comparable to the Advantage/Expectation factor which is included in the E-model for VoIP. We must however remark that a gaming experiment differs from VoIP user tests in the sense that a test user must have a certain

gaming background in order to successively participate in the experiment. The complexity of First Person Shooters may otherwise be too overwhelming and may lead to irrelevant results or even nausea. That is why we have chosen to invite two experienced user groups: experts and super-experts. The experts have a rich background in online gaming (including FPS experience), but have always mainly played for fun. The invited super-experts obviously also have plenty of experience with online gaming, but additionally have competed in various competitions.



Figure 16: Gaming experiment impression

During our two test sessions, six gamers simultaneously played Quake IV in a Free-For-All setting (Figure 16). The FFA setting is one of the more popular FPS game modes in which players constantly engage in battles to collect the most frags (kills) given a certain time limit. This mode is particularly suited for our experiment, because it is very combat oriented in comparison to for example team games, which are heavier on strategic aspects. Per test group all gamers were of the same user class in order to keep the field of competition somewhat leveled. Otherwise, large differences in skill could lead to reduced motivation and results. The delay, jitter, and packet loss of the network were altered with a network emulator, which is more thoroughly described in the setup section. Each resulting scenario lasted five minutes. After these five minutes of gameplay, the gamers were asked to give their opinion of the gaming quality. They could select one of the following five opinion scores: 5: *Excellent gaming quality*, 4: *Good gaming quality*, 3: *Fair gaming quality*, 2: *Poor gaming quality*, 1: *Bad gaming quality*

Quake IV Gaming Experiment - Scoreformulier

Scoreformulier (omcirkel per scenario wat van toepassing is)

NAAM: SVEN SPRACHT ALIAS: VICIOUS PC NR: 5

Scenario	Opinie Score*	Acceptabel**	Dominante Factor(en)***
1	1 (2) / 3 / 4 / 5	JA/NEE	Ping/Jitter/Loss/N.v.t.
2	1 / 2 / 3 / 4 (4) / 5	JA/NEE	Ping/Jitter/Loss/N.v.t.
3	1 / 2 / 3 / 4 / 5 (5)	JA/NEE	Ping/Jitter/Loss/N.v.t.
4	1 (2) / 3 / 4 / 5	JA/NEE	Ping/Jitter/Loss/N.v.t.

Figure 17: Experiment scorecard

Per scenario gamers were also asked whether they thought the quality of the test scenario was acceptable if it was played on an Internet server. Our third scenario specific question was which of the factors delay, jitter, or packet loss had the most significant effect on their opinion score. During the experiment the gaming server kept track of all kills and deaths of the participating players. After each scenario these stats were logged, with the purpose of identifying a relation between these objective performance measures and the chosen impairment factors.

Each user group had to assess a total of 33 scenarios. The 33 scenarios (which can be found in Appendix B) include both network impairment caused by a single factor, as well as impairment consisting of combinations of ping, jitter, and loss. Prior to the start of the experiment we introduced three reference scenarios corresponding to “Excellent gaming quality”, “Fair gaming quality”, and “Bad gaming quality”. We have already mentioned the fact that a visible ping value itself can influence the player’s performance and joy. For that reason the gamers were not allowed to use the scoreboard function of Quake IV, which shows the current ping.

5.4.1 Setup

The gaming test setup consisted of 6 client PC’s which were all connected to the gaming server through a Gigabit switch (Figure 18). Both the gaming clients and the dedicated server were running the latest version of Quake IV (v1.1) plus a modification called X-Battle (v0.22) [33]. This modification is used in a large number of high profile Quake IV leagues and competitions. It enables gamers and administrators to customize Quake IV to their specific needs and addresses a lot of bugs which are still present in the original version of the game.

All of the scenarios were played on the map q4dm7: Over the edge. It is a medium sized map which perfectly suits our 6 player Free for All setting. We have chosen to use the same map for the entire experiment, because otherwise our objective performance measures would not be comparable between scenarios.

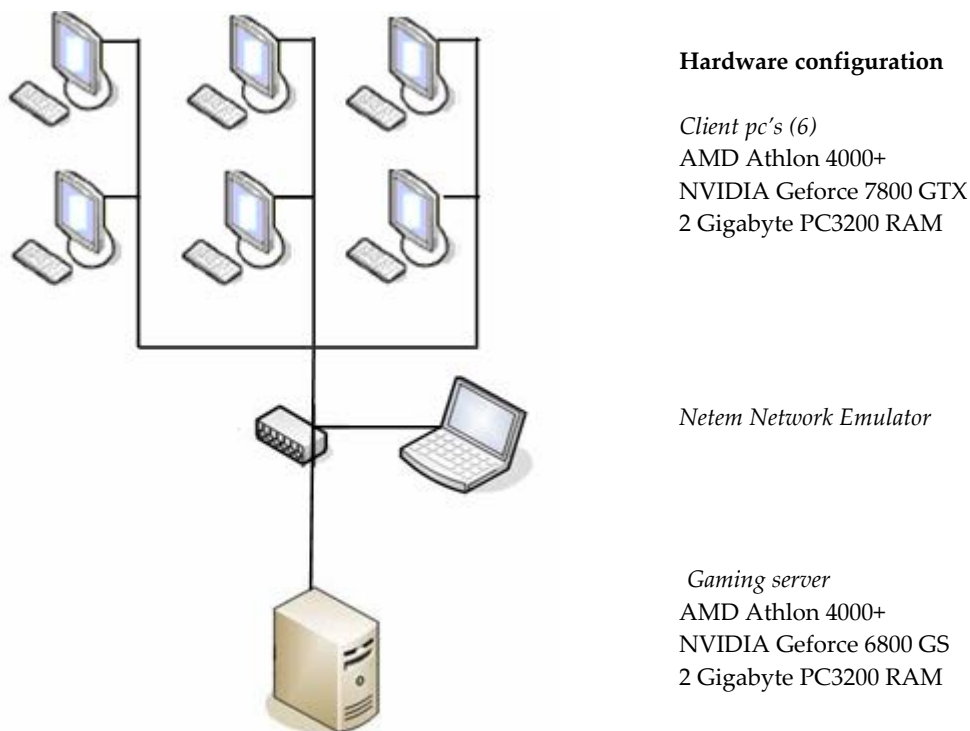


Figure 18: Gaming experiment setup

The network emulator Netem was used in order to vary ping, jitter, and loss. Netem is by default enabled in the kernel of the Fedora Linux distribution 2.6. The laptop running Fedora 2.6 was placed between the switch and the server. Both the client PC's and the server computer were equipped with last generation hardware in order to minimize the probability of performance drops. It is common knowledge that reduced hardware performance can negatively influence the subjective quality experience as well as the objective performance measures.

5.4.2 Questionnaires

Before the actual experiment started, all of the gamers had to answer a couple of general questions regarding their gaming background and habits. This information was mainly used to verify whether we invited the right gamers for the right day.

General questions

1. For how many years have you played online computer games?
2. What is your favorite multiplayer genre (FPS, MMORPG, RTS)?
3. Have you played Quake IV over the Internet, before this experiment?
 4. Have you ever competed in online competitions?
 5. What kind of Internet connection do you have?
6. When you want to play online, do you use a game browser to find a server?
 7. What range of ping times do you consider to be acceptable?
8. How often do you check your in-game ping during a normal gaming session?
9. Have you ever bought new hardware because you needed more frames per second (fps)?

As mentioned earlier, after each of the 33 scenarios the gamers answered three questions concerning the perceived gaming quality, which are stated as follows:

Scenario specific questions

1. Rate the quality of the test scenario on a scale of 1 to 5: (5) Excellent gaming quality, (4) Good gaming quality, (3) Fair gaming quality, (2) Poor gaming quality, (1) Bad gaming quality
 2. Was the quality of this test scenario acceptable to you? Yes / No
3. Which of the following factors was the most degrading factor of your gaming experience?
Ping / Jitter/ Packet loss / Not Applicable

5.5 Results

In this section we will analyze the data which we collected from the gaming experiments. We will particularly focus on answering our initial research questions from Section 1.1.

A first look at the experiment data shows us that the subjective opinions of all the participating players have a high correlation (between 0.74 and 0.95) with the MOS per scenario. This indicates that all of our test users were able to provide more or less consistent opinion scores. Nevertheless a couple of outliers were found in our data set. Two scenarios which had little to no network impairment received an opinion score of 1 (bad gaming quality), while the users in question did find the quality of these scenarios acceptable. During the experiment we accidentally observed a test user mixing up the rating 1 and 5 of our opinion scale. We have therefore chosen to eliminate these inconsistent opinion scores from our experiment data.

In the setup of our experiment we originally discriminated between two different types of gamers: experts and super-experts. We expected that players who primarily play for fun or are new to gaming, would be less concerned with the quality of their connection than players who for example compete in online competitions or are part of organized gaming clans. This expectation is confirmed by the results of our gaming experiment which show that hardcore gamers on average gave a MOS of 2.77, while the casual gamers awarded a mean MOS of 3.2. Further analysis of the experiment data however shows that it is not necessary to split up the data in order to quantify the impact of network impairment on MOS and the number of kills. The coefficients of determination of our models hardly change if we throw all our data on one pile. We have therefore chosen to use the combined data set of both gamer types in the remainder of this results section. This decision was supported by the fact that this approach increases the applicability of our models to a broader population of FPS gamers.

It is now time to start with the analysis of our impairment factors. In this factor evaluation we isolate and analyze the scenarios in which only one impairment factor is present.

5.5.1 Factor analysis

5.5.2 Regression Analysis

5.5.3 G-model

6 Conclusions and Recommendations

In this chapter we will review and summarize the most important results of our research project. The chapter will end with a number of recommendations for future research.

6.1 Conclusions

Let us recall from the problem description in Section 1.1 that the goal of this research project was to provide insight into the estimation of ping times and into the quality experience of gamers. In order to provide this insight the accompanying research questions from the same section are answered.

Part I: Gaming Performance Model

In “Part I: Gaming Performance Model” the influence of different network parameters on FPS ping times was studied. We followed a mathematical approach which heavily relied on queuing theory. The ultimate result was a Mathcad implementation which enabled us to calculate delay quantiles for different network scenarios.

In what way does interfering voice traffic influence ping times of online gamers?

We were particularly interested in the influence of VoIP traffic running over the same network. Based on our results we conclude that VoIP traffic can lead to both a decrease as well as an increase of ping times, compared to the situation in which only gaming traffic runs over a network. For our studied scenarios the addition of voice packets above 25% of total traffic leads to a relative decrease in ping times, while smaller fractions of additional voice packets result in an increase in ping times.

What is the influence of different network parameters on estimated ping times?

Other parameters of interest were the load on the aggregation link, the inter-arrival time of gaming bursts, and the Erlang order K of mean burst sizes. The first two mentioned parameters have a negative impact on the network quality of the FPS gamers: ping quantiles are significantly higher for increased parameter values. Increasing the Erlang order K has the opposite effect: the decrease in variation of the mean burst size results in less congestion in the system. Of these three parameters especially the load and the inter-arrival time of gaming bursts are important for gaming service providers. The load on the aggregation link can be directly influenced by changing the maximum number of users which are granted access to the gaming or VoIP services. We already mentioned the fact that gaming server administrators can manually set the rate at which the server sends updates back to the client. Based on our research results we recommend servers administrators to pick the lowest possible value in the range of 20-50 ms (50 is default in most FPS games). Lowering the inter-arrival times of server update packets results in a fluid responsive gaming experience, which is also reflected in the estimated delay of our performance model.

Part II: G-model

In “Part II: G-model” we focused on the influence of network impairment on subjective and objective performance measures. We conducted a user experiment to quantify this impact and to develop the Quake IV G-model.

Which factors are responsible for influencing the quality experience of online gamers?

Based on our literature study we chose to examine the impact of ping, jitter, and packet loss. Our experiment results demonstrate that ping and jitter have a significant negative effect on both the subjective and objective gaming quality, while packet loss goes unnoticed for values up to 40%.

What is the relation between the perceived quality of gamers and these factors?

Making use of a multi-dimensional regression analysis we have come up with the Quake IV G-model which enables us to predict a gamer's quality rating (expressed in MOS) based on ping and jitter values. A follow-up validation experiment showed that the model is very accurate in estimating these MOS values.

Can we identify a relation between an objective performance measure and these factors?

A similar regression model was constructed to relate the network impairment to the objective performance measure kills. Although the coefficient of determination R^2 of this model is high as well (0.895), it is not at all appropriate for estimation purposes. This is the consequence of the large impact of other factors like the number of players, the size of the map and the skill level of the players. This lack of generalization power is reflected by the validation experiment. Concerning the objective performance of gamers, we therefore conclude that although a player's score is significantly affected by ping and jitter, interpretation of an objective measure value is very hard because of the additional dependence on other factors.

6.2 Recommendations

Although the results of our research project are more than satisfactory, there is of course still room for improvement. Both our developed models are based on the characteristics of FPS games. The choice for FPS games was underpinned by research that showed that this multiplayer genre has very strict network requirements. It remains to be seen if our research results can be extended to other multiplayer games and maybe even to other platforms (e.g., mobile phones).

The quality of the Gaming Performance Model heavily relies on the quality of the underlying traffic source model. Because we were particularly interested in the tail behavior of the traffic characteristics, we could only use the traffic source model from [5]. The limited size of usable traffic measurements is the main reason why we have chosen to consider different values of the Erlang order K . More extensive traffic measurements could substantially improve the quality of our model. During the construction of our queuing model we encountered more problems, especially in the downstream direction. To determine the delay distribution of batches we had to resort to a scaled M/G/1 approximation based on simulation results of the original G/G/1 queue. This method considerably decreased the generalization power of our performance model. Follow-up research on the behavior of this complex G/G/1 queue could prove to be valuable and is therefore recommended for future work.

In the results section of Chapter 5 we already raised the question whether relating the subjective quality experience to a-priori network impairment was the most practical approach. The development of an extended G-model, which relates the measured ping and jitter values to MOS, could be even more useful for gaming service providers. This method would enable one to predict the subjective quality of Quake IV, based on some relatively short measurements in the company's network.

References

- [1] J. G. Beerends and J. M. van Vugt (2005), "A data chirp measurement method for assessing end-to-end quality of data links", White contribution COM 12-C11 to ITU-T Study Group 12, Delft, The Netherlands
- [2] T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, E. Agu (2004), "The Effect of Loss and Latency on User Performance in Unreal Tournament 2003[®]", SIGCOMM04, Oregon, U.S.A.
- [3] M.S. Borella (1999), "Source Models of Network Game Traffic", Networld & Interop. '99 Engineers Conf
- [4] D. De Vleeschauwer, M.J.C. Büchli, A. Van Moffaert, R.E. Kooij (2002), "End-to-End Queuing Delay Assessment in Multi-service IP Networks", Journal of Statistical Computation and Simulation, vol. 72, no. 10
- [5] N. Degrande, R.E. Kooij, M. Mandjes, D. De Vleeschauwer (2006, *awaiting publication*), "Modeling Ping times in First Person Shooter games"
- [6] M. Dick, O. Wellnitz, L. Wolf (2005), "Analysis of factors affecting Players' Performance and Perception in Multiplayer Games", NetGames'05, Hawthorne (NY), U.S.A.
- [7] J. Färber (2002), "Network Game Traffic Modeling", NetGames'02, Braunschweig, Germany
- [8] ITU-T Recommendation G.107 (2003), "The E-model, a Computational Model for Use in Transmission Planning"
- [9] T. Jehaes, D. de Vleeschauwer, T. Coppens, B. van Doorselaer, E. Deckers, W. Naudts, K. Spruyt, R. Smets (2003), "Access Network Delay in Networked Games", NetGames'03, Redwood City (California), U.S.A.
- [10] T. Henderson, S. Bhatti (2003), "Networked Games – a QoS-sensitive application for QoS-insensitive users?", ACM SIGCOMM 2003 Workshops, Aug.25 & 27, Karlsruhe, Germany
- [11] L. Kleinrock (1975), "Queuing Systems Volume 1: Theory", John Wiley & Sons, New York, U.S.A.
- [12] M. Oliveira, T. Henderson (2003), "What Online Gamers Really Think of the Internet?", NetGames'03, Redwood City (California), U.S.A.
- [13] P. Quax, P. Monsieurs, W. Lamotte, D. De Vleeschauwer, N. Degrande (2004), "Objective and Subjective Evaluation of the Influence of Small Amounts of Delay and Jitter on a First Person Shooter Game", NetGames'04, Portland (OR), U.S.A.
- [14] R.N. Queija (1999), "Processor-Sharing Models for Integrated-Services Networks", The Netherlands
- [15] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, L. Cottrell (2003), "pathChirp: Efficient Available Bandwidth Estimation for Network Paths", Paper 3824, Passive and Active Workshop, La Jolla (California), U.S.A.
- [16] J. Roberts, U. Mocchi, J. Virtamo (1996), "Broadband Network Teletraffic - Performance Evaluation and Design of Broadband Multiservice Networks", Final Report of Action COST 242. Springer, Berlin, Germany

- [17] C. Schaefer, T. Enderes, H. Ritter, M. Zitterbart (2002), "Subjective Quality Assessment for Multiplayer Real-Time Games", NetGames'02, Braunschweig, Germany
- [18] N. Shah, C. Haigh (2005), "The video game industry, An Industry Analysis, from a VC perspective", Hanover (NH), U.S.A.
- [19] H. Tijms (2002), "Operationele Analyse, een inleiding in modellen en methoden", The Netherlands
- [20] Ubicom (2005), "OPScore, or Online Playability Score: A Metric for Playability of Online Games with Network Impairments"

Webpages (as of May 2006)

- [21] <http://www.camiant.com/perspectives/gaming.htm>
- [22] <http://www.emerce.nl/nieuws.jsp?id=1102236>
- [23] <http://www.idsoftware.com/>
- [24] <http://www.instat.com>
- [25] <http://www.ravensoft.com/>
- [26] <http://www.theopl.com>
- [27] <http://www.theesa.com/>
- [28] <http://www.tno.nl>
- [29] <http://www.vedrashko.com/thesis/proposal.htm>
- [30] <http://www.wikipedia.org/>
- [31] <http://www.wired.com/wired/archive/14.04/wright.html>
- [32] <http://www.worldcybergames.com>
- [33] <http://www.xbattle.de/>

Television

- [34] <http://www.vpro.nl/programma/detoekomst/afleveringen/26025012/>
- [35] <http://www.vpro.nl/programma/detoekomst/afleveringen/27029984/>

Appendix A: Mathcad calculation sheets

PARAMETERS				
Tolerance	imaginary unit	max iterations	max quantile [ms]	
$TOL := 10^{-10}$	$i := \sqrt{-1}$	$L := 1000$	$Q_{\max} := 5000$	
INPUT (G = Gaming V = VoIP)				
$P := 10^{-3}$			Quantile [ms]	
$K := 9$			Erlang order	+
$N_g := 144$	$N_v := 12.5$		No. users	
Uplink (Client to Server)				
$R_u := 5000$			aggregation link rate [kb/s]	
$S_{g_up} := 80$	$S_{v_up} := 160$		Packet size [byte]	
$D_{g_up} := 40$	$D_{v_up} := 40$		Inter Arrival Time [ms]	
Downlink (Server to Client)				
$R_d := 5000$			aggregation link rate [kb/s]	
$S_{g_down} := 125$	$S_{v_down} := 160$		Packet size [byte]	
$D_{g_down} := 40$	$D_{v_down} := 40$		Inter Arrival Time [ms]	
$B_{g_up} := \frac{S_{g_up}^8}{R_u}$	$B_{v_up} := \frac{S_{v_up}^8}{R_u}$	Service time Uplink [ms]	$B_{g_up} = 0.128$	$B_{v_up} = 0.256$
$B_{g_down} := N_g \frac{S_{g_down}^8}{R_d}$	$B_{v_down} := \frac{S_{v_down}^8}{R_d}$	Service time Downlink [ms]	$B_{g_down} = 28.8$	$B_{v_down} = 0.256$
$A_1 := \frac{B_{g_up}}{\min(B_{g_up}, B_{v_up}, B_{g_down}, B_{v_down})}$	$A_2 := \frac{B_{v_up}}{\min(B_{g_up}, B_{v_up}, B_{g_down}, B_{v_down})}$		$A_1 = 1$	$A_2 = 2$
$A_3 := \frac{B_{g_down}}{\min(B_{g_up}, B_{v_up}, B_{g_down}, B_{v_down})}$	$A_4 := \frac{B_{v_down}}{\min(B_{g_up}, B_{v_up}, B_{g_down}, B_{v_down})}$		$A_3 = 225$	$A_4 = 2$

UPLINK

$$P_1 := \frac{\frac{N_g}{D_{g_up}}}{\frac{N_g}{D_{g_up}} + \frac{N_v}{D_{v_up}}} \quad P_2 := \frac{\frac{N_v}{D_{v_up}}}{\frac{N_g}{D_{g_up}} + \frac{N_v}{D_{v_up}}} \quad P_1 = 0.92 \quad P_2 = 0.08$$

$$\text{bitrate}_g := N_g \cdot \delta \cdot \frac{S_{g_up}}{D_{g_up}} \quad \text{bitrate}_v := N_v \cdot \delta \cdot \frac{S_{v_up}}{D_{v_up}} \quad \text{bitrate}_g = 2.304 \times 10^3 \quad \text{bitrate}_v = 400 \quad [\text{kbit/s}]$$

$$\rho_u := \frac{\text{bitrate}_g + \text{bitrate}_v}{R_u} \quad \rho_u = 0.541 \quad \text{Uplink Load}$$

$$\lambda := \text{root}\left[p \cdot (P_1 \cdot A_1 + P_2 \cdot A_2) - \rho_u \cdot (P_1 \cdot \exp(p \cdot A_1) + P_2 \cdot \exp(p \cdot A_2)) - 1, p, 0.00001, 100\right] \quad \lambda = 0.939$$

$$R := \frac{1 - \rho_u}{\rho_u \cdot \frac{P_1 \cdot A_1 \cdot \exp(\lambda \cdot A_1) + P_2 \cdot A_2 \cdot \exp(\lambda \cdot A_2)}{P_1 \cdot A_1 + P_2 \cdot A_2} - 1} \quad R = 0.655$$

$$F_{du}(t) := R \cdot \exp(-\lambda \cdot t)$$

$$Q_{du} := \text{root}(F_{du}(t) - P, t, 0, Q_{max}) \cdot \min(B_{g_up}, B_{v_up}, B_{g_down}, B_{v_down}) \quad Q_{du} = 0.884 \quad \text{quantile [ms]}$$

DOWNLINK PACKET DELAY WITHIN BURST

$$E(x, k) := \left(\sum_{i=1}^k \frac{x^{i-1}}{(i-1)!} \right) \exp(-x) \quad \text{TDF of Erlang distribution}$$

$$F_{dp}(t) := \frac{1}{K-1} \cdot \sum_{k=0}^{K-2} E(\beta \cdot t, K - k - 1)$$

$$Q_{dp} := \text{root}(F_{dp}(t) - P, t, 0, Q_{max}) \cdot \min(B_{g_up}, B_{v_up}, B_{g_down}, B_{v_down}) \quad Q_{dp} = 54.643 \quad \text{quantile [ms]}$$

DOWNLINK BURST DELAY

$$A_3 = 225 \quad A_4 = 2$$

$$\text{bitrate}_g := N_g \cdot \frac{S_{g_down}}{D_{g_down}} \quad \text{bitrate}_v := N_v \cdot \frac{S_{v_down}}{D_{v_down}} \quad \text{bitrate}_g = 3.6 \times 10^3 \quad \text{bitrate}_v = 400 \quad [\text{kbit/s}]$$

$$P_1 := \frac{\frac{1}{D_{g_down}}}{\frac{1}{D_{g_down}} + \frac{N_v}{D_{v_down}}} \quad P_2 := \frac{\frac{N_v}{D_{v_down}}}{\frac{1}{D_{g_down}} + \frac{N_v}{D_{v_down}}} \quad P_1 = 0.074 \quad P_2 = 0.926$$

$$\Delta := \frac{\text{bitrate}_v}{\text{bitrate}_g + \text{bitrate}_v} \quad \Delta = 0.1 \quad \text{Fraction VoIP traffic}$$

$$\rho_d := \frac{\text{bitrate}_g + \text{bitrate}_v}{R_d} \quad \rho_d = 0.8 \quad \text{Downlink Load}$$

$$\beta := \frac{K}{A_3} \quad \beta = 0.04$$

$$\gamma := \text{root} \left[\rho_d \left[P_1 \frac{\beta^K}{(\beta - \rho)^K} + P_2 \cdot \exp(A_4 \rho) - 1 \right] - \rho \left(P_1 \frac{K}{\beta} + A_4 P_2 \right), \rho, 0.00000001, \beta - 0.00000001 \right]$$

$$C := \frac{1 - \rho_d}{\rho_d \frac{P_1 \frac{K \cdot \beta^K}{(\beta - \gamma)^{K+1}} + A_4 P_2 \cdot \exp(A_4 \gamma)}{P_1 \frac{K}{\beta} + A_4 P_2} - 1} \quad C = 0.84$$

$$S_{\text{tmp}} := 5.792 \cdot \rho_d + 0.115 \cdot K - 3.172 \cdot \Delta \quad S_{\text{tmp}} = 5.351$$

$$S := \text{if}(S_{\text{tmp}} < 1, 1, S_{\text{tmp}}) \quad S = 5.351 \quad \text{Scaling factor}$$

$$\gamma := \gamma \cdot S$$

$$F_{\text{db}}(t) := C \cdot \exp(-\gamma t)$$

$$Q_{\text{db}} := \text{root}(F_{\text{db}}(t) - P, t, 0, Q_{\text{max}}) \cdot \min(B_{g_up}, B_{v_up}, B_{v_down}, B_{g_down}) \quad Q_{\text{db}} = 86.819 \quad \text{quantile [ms]}$$

TOTAL DELAY

$$d_u(s, k) := \text{if}(k = 0, 1 - R, 0) + R \cdot \frac{\lambda}{(\lambda - s)^{k+1}} \quad \text{This is } D_u^{(k)}(s)/k!$$

$$w(s, k) := \text{if}(k = 0, 1 - C, 0) + C \cdot \frac{\gamma}{(\gamma - s)^{k+1}} \quad \text{This is } W^{(k)}(s)/k!$$

$$p(s) := \frac{1}{K-1} \cdot \sum_{k=0}^{K-2} \left(\frac{\beta}{\beta - s} \right)^{K-1-k} \quad \text{This is } P(s)$$

$$F_{\text{tot}}(t) := w(\lambda, 0) \cdot p(\lambda) \cdot R \cdot \exp(-\lambda t) + d_u(\gamma, 0) \cdot p(\gamma) \cdot C \cdot \exp(-\gamma t) + \frac{1}{K-1} \cdot \sum_{k=0}^{K-2} \left[\sum_{l=0}^k (-\beta)^l \cdot \sum_{m=0}^1 d_u(\beta, m) \cdot w(\beta, 1-m) \right] \cdot E(\beta t, K-k-1)$$

$$Q_{\text{tot}} := \text{root}(F_{\text{tot}}(t) - P, t, 0, Q_{\text{max}}) \cdot \min(B_{g_up}, B_{v_up}, B_{g_down}, B_{v_down}) \quad Q_{\text{tot}} = 106.103 \quad \text{quantile [ms]}$$

Appendix B: Experiment scenarios

SCENARIO	NETEM ref.	Ping	Jitter Pareto	Jitter flap	Packet Loss
19	1	20	0	0	0
13	2	40	0	0	0
27	3	80	0	0	0
31	4	160	0	0	0
8	5	320	0	0	0
21	6	0	10	0	0
32	7	0	20	0	0
14	8	0	40	0	0
18	9	0	80	0	0
7	10	0	160	0	0
23	11	0	0	10	0
9	12	0	0	20	0
2	13	0	0	40	0
15	14	0	0	80	0
29	15	0	0	160	0
16	16	0	0	0	2
3	17	0	0	0	5
28	18	0	0	0	10
25	19	0	0	0	20
6	20	20	20	0	0
30	21	40	40	0	0
26	22	60	60	0	0
22	23	80	80	0	0
24	24	160	160	0	0
11	25	320	80	0	0
10	26	80	20	0	5
17	27	160	40	0	10
1	28	80	10	0	20
20	29	320	40	0	5
4	30	160	20	0	10
12	31	100	20	0	20
5	32	0	0	0	0
33	19b	0	0	0	40

Appendix C: Additional graphs