Network Gaming

Performance and Traffic Modeling

MATTIAS ÅKERVIK

KTH Information and Communication Technology

Master of Science Thesis
Stockholm, Sweden 2006

COS/CCS 2006-17

Network Gaming: Performance and Traffic Modeling

Mattias Åkervik

Thursday, November 30, 2006

Stockholm, Sweden

Abstract

There are several different types of games that are played in multiplayer mode over networks. The type of network games that, from a network's perspective, are the most demanding is real-time based multiplayer games. Users of such games both assume and require that game play interaction happens in near real-time and these games often support a large number of simultaneous players. Most networks are specialized to either voice traffic (such as the first and second generation of mobile networks) or data traffic (such as wired data networks). It is not clear that the requirements for such real time games can always be met on either type of network. The core of this thesis investigates the performance requirements real-time multiplayer games place on packet switched data networks and the connection between network impairments and game quality degradation. Traffic generated by network games distinguishes itself from other traffic both regarding its general characteristics and the requirements it places on the network. Understanding these traffic characteristics, requirements, and what consequences failures to support such requirements entail are of great importance when designing new networks in order to guarantee suitable quality of service for such real-time games.

Keywords: Network performance, Online gaming, Network gaming requirements, Game quality

Abstract in Swedish

Det finns idag en stor mängd datorspel som spelas i flerspelarläge över nätverk. De spel som från ett hårdvaru- och nätverksperspektiv ställer högst krav är realtidsbaserade flerspelarspel. Slutanvändare av dessa realtidsspel både förutsätter och tar för givet att interaktionen sker i så nära realtid som möjligt samtidigt som dessa spel ofta stödjer ett stort antal samtidiga användare. De flesta nätverk är i första hand anpassade för rösttrafik (som första och andra generationens mobilnät) eller datatrafik (som trådade datanätverk). Det står inte klart huruvida någon av dessa nätverk kan garantera tillräcklig prestanda för att en acceptabel spelkvalité skall uppnås för slutanvändaren. Kärnan i denna rapport utreder vilka krav som dagens mest krävande realtidsbaserade flerspelarspel ställer på nätverken de spelas över samt kopplingen mellan brister i dessa nätverk och den upplevda spelkvalitén för användarna. Trafik genererad av realtidsbaserade flerspelarspel särskiljer sig från annan trafik både när det gäller generell karaktäristik och när det gäller de krav som de ställer på nätverken. Det är viktigt att ha en förståelse kring den trafik som dessa spel skapar, kraven de ställer, samt de konsekvenser ett misslyckande av upprätthållande av sådana krav medför. Denna förståelse är av yttersta vikt när man designar nya nätverk för att kunna erbjuda en passande Quality of Service för denna typ av interaktiva multimediatjänster.

Acknowledgements

First I want to thank my supervisor at KTH, Prof. Gerald Q. Maguire, for his fast comments and guidance. I want to thank my supervisor at Ericsson, Tord Westholm, for helping me structure my work, for his valuable comments, and for giving me free hands forming this thesis. Then I also want to thank Carl-Gunnar Perntz for actively supporting my work during the whole time. I also want to express my gratitude to all employees at Ericsson that have supported and helped me with problems I was encountered with along the way. Finally I want to thank everyone that participated in the game quality survey.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Until recently telecommunication networks have been optimized only for voice communication. The wired network infrastructure, as used by Internet, is optimized for data transfer. Wireless wide area cellular networks were also initially optimized for voice communication, but are also moving more and more towards data communications. Today data traffic appears in mobile networks and voice traffic in the wired packet switched networks (e.g. Voice over IP). Formerly handsets for mobile networks were voice-centric, but today more advanced cell phones have significant support for data due to the advanced services data makes possible. As a result of this development data-traffic's share of the next generation of mobile networks is predicted to rise. According to several sources [22, 29, 30] the mobile gaming revenues in North America will be worth $1200 million and for Western Europe $2700 million. The fastest growth of online gaming is predicted to take place in Asia and the Pacific [22]. It is not only the gaming industries that make profit from mobile gaming; it will also drive transport revenues for mobile carriers and create a need for gaming enabled terminals.

Traffic generated by online games generally distinguishes itself from other Internet traffic today both in its characteristics and in its requirements on the network. In some ways game traffic has similar requirements to voice traffic, but will still have different traffic characteristics.

Since the game traffic characteristics differs between the different game genres it is not a trivial task to come up with a general traffic model that could be applied to all existing or up-coming games. Network impairments that are relevant to all types of online games are high latency, packet loss, and reordered packets. These problems can, and most certainly will, once the impairments reach a certain point, negatively impact the end user's perception of the game quality. Since in the end, the crucial issue is the end user's perception of the gaming experience, it is of great interest to avoid exceeding these thresholds. This thesis will only study a specific, carefully chosen, genre of games that have the highest performance demands on the network, i.e. real-time multiplayer games.

A key question is if online gaming popularity will continue to grow given today's "low-performance" wide area cellular network(s).

1.1 Problem statement

Today’s wide area cellular access networks limit game quality for the end user. Since the terminals and network infrastructures available today were not developed with mobile gaming in mind, high latency is not a significant problem for the games played on mobile telephones available today. This latency will not be a major obstacle for the gaming industry until end user game terminals have enough performance to run more complex real time multiplayer games. Due to the current development of the hardware components this will most probably happen within the coming four years (before year 2010). To enable online gaming through these terminals, it is of great importance that mobile networks are designed and operated with a good understanding of what requirements games may put on the network. This not only applies to games played on our cellular telephone handsets, but also traffic from portable game consoles or other personal computers that may use wide area cellular networks to connect to online gaming services, which is the case with mobile broadband.

1.2 Thesis objectives

- Analyze factors affecting network gaming performance by describing what requirements various types of network games put on terminals, networks, and services. Included is also an assessment of what the minimum performance requirements are for a cellular network in order to support such gaming.
- Construct a network gaming traffic model based on, preferably, several traffic traces from representative network games. This includes an appropriate model general enough to carry over to any popular network game, along with parameter values based on the specific traces analyzed.
- A study of the network gaming market including the size (number of players and revenues), structure, and if possible how to classify network gamers (i.e., are these distinct classes of gamers). What is the willingness of each class to pay for network gaming? Is this class strong enough to drive network performance demands?

1.3 Related work

Gaming in general and network gaming in particular embraces a wide variety of research topics. There exist studies related to gaming ranging from artificial intelligence (as implementation in games) to how to optimize multiplayer gaming protocols [13]. This may not be so surprising since a successful online game contains components from all these areas. Studies in the area of multiplayer gaming are mainly performed by commercial gaming companies, academic institutions, or the military. The United States Defense Department annually spends $1.5 billion on modeling and simulation [1]. Many of the areas where the military spends money, i.e. developing new models or simulation techniques (simulating war fighting situations, evaluating new tactics, etc.) are directly applicable to gaming in virtual worlds.

In the specific area of interest for the scope of this thesis, there have only been a few earlier studies. Although there have been extensive studies of network gaming in wired packet switched networks [16, 15] (where most studies have been carried out concerning the popular game "Counter strike" [15, 23]) but when it comes to online gaming over wireless links the studies have had a more limited scope - typically examining only simple network games (simple from a network perspective) playable using a java virtual machine (J2ME™) on a mobile telephone [18]. There also exist some studies on how to ensure high end user game quality based on an approach to provide quality of service (QoS) for mobile gaming [14]. There exist online gaming market forecasts from both a business and economic aspect [22, 29, 30].

1.4 The reader

The reader of this report is assumed to have general knowledge about computer networks and TCP/IP in particular. Knowledge about wireless networks and their limitations will help the user understand the problems that exist today with online gaming over such networks. It is not needed, nor expected, that the reader have any prior experience with online gaming, although earlier practical experience with online gaming can help the readers understand how the different impairments are perceived by an end user.

1.5 Overview

This thesis is divided into eight main chapters. The first chapter contains an overview of the disposition of the thesis, the problem statement, the thesis objectives, and a description of the intended reader. Chapters two and three will give an orientation to network games in general and network gaming from both a technical and social view, including a picture of the end users of gaming services. In chapter four, technical challenges is covered, especially the kind of problems that exist in today's networks and there impact on the game quality of today's games. The following two chapters will look closely at the traffic online games produce and discuss some laboratory experiments together with a general traffic model. This model is designed to be general enough to cover highly demanding multiplayer interactive network games. Chapter five examines games in wired packet-switched networks, while chapter six considers gaming in cellular networks. The following chapter examines network gaming from economic and business viewpoints. Including a cursory investigation of the online gaming market and the potential revenues for network infrastructure vendors, gaming industry, and network operators. The last chapter will conclude what has been discussed throughout the thesis regarding what requirements future online network gaming possibly may place on tomorrow’s networks and how the development of online gaming may look in the coming 5 years.

Chapter 2
Network games

Only a subset of all (released) games has network gaming functionality. In the past most games were designed for a single player, sometimes with score reporting for comparison with other players' scores. With the growth of Internet and the growth in broadband home connectivity, user demand for online multiplayer games has increased. Today, most new games include some kind of online multiplayer functionality, an attractive feature that is popular among gamers today.

From a network perspective different types of games have different requirements. Some literature shows that the requirements are rather homogeneous *within* each genre. In this chapter different kinds of games and game genres, in which network gaming are common, is examined and explained. The objective for this chapter is to give an introduction to the specific games and genres that are closer studied in later chapters.

2.1 Game genres

Dividing games into genres can be done in more than one way. In this report I have chosen to divide them with respect to how the games are played. This strongly reflects what requirements they put on the network when it comes to online gaming.

2.1.1 First Person Shooter

First person shooters (FPS) are games where you see the world from a character's eyes. Although it is **not** a requirement that these games actually include some kind of shooting, but in fact most of them do. It is a common opinion that the first FPS, and the ancestor of today's FPS games, was Wolfenstein 3D, released in 1992 by ID Software. Games belonging to the FPS genre are often aimed at online gaming. Most, if not all, include some sort of multiplayer functionality. A few examples of successful FPS games are the Quake series containing four games and Halflife, which perhaps is more known for its modification "Counter Strike". The later has been played actively since it came in November 2000 [33] and is still today popular (even if the graphical engine has been upgraded since the first release).

2.1.2 Third Person Shooter

Third person shooter (TPS) games have many things in common with FPS. The difference is that in a third person shooter game you follow the character from a third person point of view. A general theme for TPS (also valid for FPS games) is that the player has to perform missions to get new equipment, weapons, keys, or information that helps the player advance to the next level or finish the game. TPS is a relatively small genre, but it includes some very popular games; such as Tomb Raider [48], Max Payne [49], and Grand Theft Auto [50].

2.1.3 Real Time Strategy

Real time strategy (RTS) games are simply strategy games that are played in real time. Here real time refers to the fact that time is ticking continuous and that at any moment you can interact with the game and this movement occurs at a specific time. A very common game play pattern in real time strategy games are:

1. Harvest resources (gold, stone, wood, oil, …)
2. Build buildings and train soldiers/units
3. Destroy your enemy with your newly build armies, buildings, ...

The general appearance of an RTS-game is that you observe the world, your troops, and your buildings from above. Subsequently you will be given the possibility to zoom in and out and to freely change your view point of the tub or three dimensional game world.

2.1.4 Turn based games

In contrast to real time games, turn based games only allow interaction with the game when it is your turn. Classic board games are good examples of turn based games. To this genre belong almost all board games and turn based strategy games, such as the classic strategy game Civilization (Microprose, 1991) where your goal is to build up an army that will remain victorious over time. If every game is taken into account, then turn based games are, if not the biggest genre, stand for a large share of all gaming activities. The main advantage of turn based games is that they often have analogies with classical board or card games, hence they are often easy to learn and often known in advance, therefore they are tempting for new player.

2.1.5 Sports games

Sports games have always been popular. Soccer, basketball, ice hockey, golf, and tennis, to name only a few sports, have all been transformed into popular computer games. Even racing games, such as the popular game series “Need for speed” [51], first version released 1994, are commonly grouped in the sport game genre. When it comes to multiplayer gaming sport games are not as actively used as previously mentioned games. One reason is that sport games often have more long term goals than the other games, i.e. win the world cup in soccer or win a tournament. Since it is troublesome to arrange large tournaments online and have many people that are able to participate at fixed times, most sport games have focused on single players.

2.1.6 Role Playing Games

The main idea that permeates role playing computer games (RPG) is that the player acts as a fictitious person in the game and together with the other players performs different kinds of missions (i.e. the players are actors in roles). In many games the story takes place in a fantasy world with human-like creatures. RPG is a popular genre among its players especially for social aspects, as you are communicating with other players, both via voice and text messages, during your mission.

2.1.6.1 Massive Multiplayer Online Role Playing Games

Massive multiplayer online role playing games (MMORPG) are role playing games for multi-user online gaming. The business idea behind MMORPG is in many ways brilliant. The developer creates and sells a game that only is playable online. To use this online service, the end user needs a (online) subscription. To attract the end user, the first month(s) of participation are usually free of charge. These kinds of games are more common in Asia, but have a strong presence even in Europe and North America. One of the more popular MMORPG game today is World of Warcraft (WOW) [58].

2.2 Platform interoperability

Generally games in all genres discussed in this chapter are available on more than one platform. In addition to games developed for J2ME (which in some sense is platform independent), games are also developed for specific target platforms. Popular games are generally ported to most platforms, so in some cases it is possible to challenge users who are using platforms other than the one you are using. One problem this inhomogeneity causes is fairness with regard to how user input is registered. The input methods of the different game terminals may vary significantly. On a portable game terminal you are often limited to input using buttons and joysticks. While with a desktop PC you can use your keyboard, mouse, or any other input device connected to the computer. Another issue is how the game is presented to the user. A desktop PC user may have an advantage as he/she has a large display consisting of millions of pixels (i.e. 1280x1024 pixels or greater) while the mobile user might be limited to only thousands pixels (often a resolution of 320x240 pixels). The user’s input possibilities are not decisive for the outcome in the game, unless the user input is tightly connected to the level of the user’s performance. A game of chess would not cause any multi platform online game problems due to the terminals, since a larger screen and higher resolution input will not give a user greater chance of winning. While theoretically it is possible for a portable terminal user to play against a PC user of the same game, in reality it is not currently common.

Whether or not we will see a convergence between PC based online game services and mobile terminal game services is not obvious, but the development of portable game terminals with better input controls, larger screens, and better feedback provide the prerequisites for a successful integration sooner rather than later.

Chapter 3
End users / Gamers

This chapter discusses a very important group of people, both for hardware vendors and service operators, called gamers (i.e. the end users of a game). These gamers bring revenue to game developers, mobile network operators, terminal vendors, and to the network infrastructure vendors. In this chapter we will look at (1) the group of people that call themselves gamers, (2) the group that do not consider themselves gamers, but still play games occasionally (casual gamers), and (3) the small group of hard core gamers that play whenever they can and wherever they can.

3.1 Gamer profile

The simple definition of a gamer is a person that plays computer games (not necessarily multiplayer games). Gamers can be divided into different classes based on how much time they spend gaming and their intentions. Traditionally most gamers were young males, but that trend is changing. Currently 38% of all gamers in America are females vs. 62% males. When it comes to online gaming 42% of gamers are females vs. 58% males according to the entertainment software association [26]. Currently the distribution of gamers is rather evenly balanced regarding both age and sex.

3.2 Types of gamers

All gamers have their own gaming desires and habits. Some gamers play for fun once a month while some play for hours each day.

3.2.1 Casual gamer

Casual gamers play occasionally and might not even think of themselves as gamers. Games are seen as something fun to do. Casual gamers are unlikely to be willing to spend significant money on gaming, but they are the largest gamer class with respect to number of individuals. For game developers that are targeting this gamer class it is of great importance to create games that are easy to get started with and easy to play.

3.2.2 Hardcore gamer

The opposite of a casual gamers is a hardcore gamer. Hardcore gamers see their gaming as a top priority and devote a large part of their leisure time to games. Since hardcore gamers put a lot of time into gaming they are often willing to spend a larger amount of money on their hobby. Their consumption of new games, content, and gaming hardware makes hardcore gamers a driving force for the gaming and related industries (such as hardware vendors, anti-cheat software developers, game service providers, and game cafes). Hardcore gamers often play a game until they become very good at playing it, then they move on to the next attractive new game title. Hardcore gamers are often members of various game specific communities and may play in teams, also called clans.

3.2.3 Competitive gamer

Competitive gamers' main intention with their gaming is the enjoyment of competing with other online gamers (although instead of pure enjoyment, the intention could be to earn money from tournaments). This type of gamer often chooses a game to play completively based upon available tournaments and the amount of prize money. Competitive gamers often specialize in a specific genre of games and chose new games within his genre, most commonly first person shooter (FPS) or real time strategy (RTS) games. It is common that competitive gamers enter tournaments together with other gamers as a team, a so called clan.

3.3 Gaming communities

It is common that communities form around popular games. Often these communities are created by group of hardcore gamers without any explicit economic goals. The goals of an online game community could be to promote fair play, create events like tourneys for community members, or hosting public gaming leagues. The number of members of a community can range from a small number of friends that are playing for fun to large communities with hundreds of thousands of members (e.g. Major League Gaming [27]). The communication between the members within a community is usually achieved via web forums, Internet Relay Chat (IRC, a form of Internet real time chat designed for many-to-many communication), or handled within the game.

3.4 Professional tournaments

Online game tournaments are not a new idea. Concurrent with the increasing value of the gaming market the prize money has also risen. Today professional gaming events are a great opportunity for gaming hardware vendors to advertise to a strong group of eager customers. The large world-wide interest has driven prizes to levels that were unthinkable just a couple of years ago. For example, the upcoming CPL 2006 Championship final which will take place in December 2006 will have a total purse of $150,000 [28]. The biggest leagues (those that have the most prize money) usually have qualification rounds around the world where competitive gamers compete for a limited number of final tickets. Then all the qualified gamers meet to determine who will win the prize [28].

3.5 Personal integrity

As a gamer you may need to enter personal information in different situations. For example, when you register your game copy to be able to play online, apply for a subscription to an online game service, or when connecting to a game server. Often you are forced to leave your full name, or e-mail address, but sometimes your birth date (so the game service provider can ensure that the user is over a specific age) or a credit card number (for billing the subscription fee). It is in most cases clear to the end user who is responsible for handling the information he/she entered (e.g. registering at the game developer's game site), but it might not be as clear when it comes to online communities or game servers.

Chapter 4
Technical aspects of online gaming

Networked gaming applications are rather complex as they include aspects from many different research areas. In order to deliver a high game quality with a satisfied end user it is important that all parts performing as planned. In this chapter a technical overview of online games will be given. Together with this overview a number of technical challenges and problems is discussed together with both existing and potential solutions.

4.1 General requirements

Computer games have been a force that has driven the computer industry towards higher performance equipment. Developers strive to develop more challenging games regarding gaming experience, graphics, and the experience of online gaming. This creates increased requirements on terminals, networks, and servers. Today, if you want to play the newest games, the performance of your two year old equipment is often no longer sufficient. You will need the latest when it comes to CPU, memory, and graphics. It is these three components that primary limit the client. When it comes to the server, the parts of most interest are the CPU, memory, and Internet connection capacity. Even if the server software does not place high requirements on the hardware, new equipment is preferred for gaming sessions containing of more than a couple players.

Problems caused by malfunctioning or slow components could in many situations negatively affect the end user, just as network impairments will. Throughout this report it is assumed that the software is running correctly on both the terminal and the game server, and we exclude any problems due to malfunctioning terminal or server hardware. Therefore all visible problems are directly a result of the current network state.

4.1.1 Server

General requirements for a dedicated game server today are:
(Typical requirements of popular PC games are given inside parentheses)

- Low latency routes to the end users.
- Large amount of memory (1GB or more).
- A fast CPU (>2GHz).

Today most servers are located at strategic places in the Internet infrastructure, such as at companies, schools, or in the home of a user with a broadband connection. Common for all three are that they all have low latency and high throughput connections to the Internet. Since the broadband connections available to end users are significantly better than 5 years ago (e.g. broadband connections with a data rate of 20-100 Mbits/second are common in Sweden), thus more and more game servers are run and hosted by end users. In general there is an interest from the game developers' to have as many server nodes of their game software running as possible. From the developers' point of view, having end users hosting their game services / game servers is advantageous since they neither have to pay for the bandwidth nor the server equipment. As a result it has lately become easy to set up a game server of your own, as the game developers provide precompiled server software and the only settings needed to configure it is often concentrated in a single file. This solution of is commonly used within the FPS genre. When it comes to other genres, it is more common that the game servers are hosted by game developers or an affiliating company.

4.1.2 Terminals

Today a large variety of gaming terminals are available to the end users. There exist portable and stationary terminals, terminals dedicated to gaming, and those where gaming is simply a subset of the terminal's functionality (e.g. a modern cellular telephone). When it comes to game quality today, the best choice of terminal for a mobile user is either a PlayStation Portable [19] or Nintendo DS Lite [20]. Both these terminals are dedicated to gaming and have embedded wireless network interfaces (IEEE 802.11b/g) to facilitate online gaming. Within a few years we will most likely see portable gaming terminals with both WLAN and cellular network interfaces enabling a new dimension of online gaming and online gaming services (specifically mobility).

A big difference between console game terminals, such as Microsoft's Xbox and Sony's PlayStation, and PC games is that the console has fixed hardware, i.e. once the console has been shipped from the factory, the game developers are forced to optimize the software for the available hardware of this console. With the PC game market the case is somewhat different. For PC hardware there are hundreds of vendors producing products that can be combined in thousands of ways to augment a PC. New PC games strive to compete against other game developers with outstanding graphical effects and fancy graphics, which induce a requirement for the latest central- and graphic- processing units. If the computer components have too low performance to support the game, the resulting impairments in game quality are sometimes similar to the game quality degradations caused by network impairments (more on this topic in section 4.5.5). In Table 1 the requirements for Battlefield 2, a representative PC game of today, could be found (as stated on the box).

Table 1. Client equipment requirements for Battlefield 2

Requirement for Battlefield 2 (as stated on the box)	
Operating system	Windows XP (32-bit)
CPU	1,7 GHz
Memory	512 MB RAM
Available hard disk space	2,3 GB
DVD-ROM	8x speed
Graphic card	128 MB with support for 1.4 Pixel Shader
Network interface	TCP/IP support

4.1.3 Network

Even though newer games exploit compensation algorithms [6, 7, 9] (see section 4.5) thus reducing the required throughput, the higher throughput of the end user access networks have made network impairments almost unnoticeable in all but the most demanding games. The most important requirement on the network infrastructure is the requirement for low round trip delays, this distinguishes game traffic requirements from requirements from most other services. The requirements differs for each genre, but generally a round trip time below 80 ms is barely noticeable, while a round trip times above 400 ms makes most games unplayable. The second most important requirement on the network concerns packet loss. A rule of thumb is that the lower the packet loss rate the better. Games are rather forgiving when it comes to packet loss; generally a few percent of the packets can be lost without disastrous consequences for the gaming experience (see Table 6). Latency and packet loss requirements are examined in chapter 5.

4.1.4 User

The user of most online game services generally has some prior gaming experience. However, this is not a formal requirement since most games include some kind of training mode where the user can learn the game and learn how the user interacts with the game. In some games there are game servers that require a specific level of user gaming skills. This is often required to ensure an even level of skill between the players on the server (and thereby more even game play). Often you have to qualify to get access to the restricted server by playing on a public server to show that your skills are sufficient for the target level of play. For the occasional gamer there exist so called “Free For All” (FFA) servers that are open for everyone without any restrictions.

A game’s ability to acclimatize a user is correlated to the user’s likelihood and willingness to continue to play the game and, in the end, the user’s interest in eventually paying for online gaming services connected to this game. More about the users of online games and social aspects can be found in chapter 2.

4.2 Genre specific requirements

The requirements, as indicated in earlier chapters, differ between different genres. The most demanding category of online games is real-time games. The most demanding and also popular subcategory of games in this category is first person shooters and real time strategy games. Even turn based games are significant, due to the fact that it is one of the most played game genres. Table *2* gives a rough estimate of the requirements each genre requires for both acceptable and good game quality. *Acceptable* denote that the game is playable, but that it may result in noticeable game quality impairments, which may not be noticeable by an inexperienced player (however, the quality may not be accepted by experienced gamers). When *Good* performance is attained, the game quality is accepted even by experienced gamers and involves little or no noticeable game quality impairments. As could also be seen with the throughput requirements, once the game is playable, the throughput has lower impact on the game quality compared to packet loss and latency. All these requirements should be seen as rough estimates of representative games from each category and are not necessarily applicable to a specific title. The packet loss rate for turn based games has a low impact on the perceived game quality of the game.

Table 2. Genre specific network requirements

Rough genre specific requirements		**First-person shooter (FPS)**	**Real-time strategy (RTS)**	**Turn based games**
Latency (RTT)	Acceptable	<120 ms	<450 ms	<4000 ms
	Good	<40 ms	<200 ms	<1000 ms
Packet loss rate	Acceptable	<8 %	<10 %	
	Good	<4 %	<5 %	
Throughput (per direction)	Acceptable	60 kbit/s	80 kbit/s	<5 kbit/s
	Good	60 kbit/s	80 kbit/s	<5 kbit/s

4.2.1 First-person shooter (FPS)

The first person shooter genre has become extremely popular in recent years (especially the game "Counter Strike" [15]). They are perhaps the most played online genre today. Important properties to an end user of a FPS-game are the user's reaction ability and the user's skill in reading the environment and the behavior of other players. This means that the real-time factor is significant, thus leading to strict latency bounds. From a network perspective FPS-games are the most demanding genre today played over networks. Generally FPS-games require a round trip time less than 120 ms to deliver a game quality that most would accept and a round trip times below 40 ms to leave game quality impairments unnoticeable (see Table *2*).

4.2.2 Real Time Strategy (RTS)

Like first person shooters, real time strategy games are a real time game where your action is based upon your enemies' positions and actions. Since real time strategy games do not depend as much on the user's reaction ability the requirements on the network are less stringent. Generally a round trip time less then around 450 ms is not noticeable. With an RTS-game the game opponent(s) election process (the process where the player searches and elect his opponent(s)) often takes longer time than a server election process for the FPS-game. In a RTS gaming session each individual player influences the outcome of the game to a greater extent than in FPS games, which makes it unpopular to quit in the middle of a game session. Generally a RTS-game session is longer than a FPS-session, thus a connection loss is more devastating in a RTS-game than in a FPS-game where a connection could easy be reconnected (unfortunately persistence is generally not good in any of the RTS or FPS game when compared to MMORPG (MMORPG games often creates a persistent universe and continue to exist through the game regardless of the players connection status)).

4.2.3 Turn based games

Turn based games are one of the oldest and perhaps most played game genres. Generally each game session lasts less then 15 minutes and due to the similarity with classic board games the learning time is minimal. Turn based games are based upon a turn based model where every end user only interacts with the game during his or her turn. Since turn based games do not have real time requirements, a round trip time of several seconds is in most cases considered acceptable. As a result of their low network requirements, turn based games will not be studied further in this report.

4.3 Network game server software architectures

The server architecture of a network game is dependent on the implementation of the game. Generally there exist a couple of architectures that are widely used. These are generally client-server and peer-to-peer, each is considered in the next sections.

4.3.1 Client-Server

The most common architecture is a centralized server that all clients connect to. In a client-server architecture most often one or more server(s) provides the game service to end users. Each client only needs to connect to the server, i.e. a client does not directly connect to other clients. In those rare cases where more than one server is used, the servers may connect to each other. The servers' role is to distribute the state of the game to all connected clients. Multi-server architectures are more commonly used within the RPG-genre. When more than one server is used, either the game environment or the users are partitioned between the servers.

4.3.2 Peer-to-Peer

The peer-to-peer model relies on every node having a connection to every other node. In general peer-to-peer architectures are either distributed or replicated. A distributed structure means that all nodes together possess all the data about the state of the game environment. In a replicated architecture **each** node manages a replica of all data. The advantage of a replicated data structure is that it is tolerant to loss of nodes at the cost of more network traffic since every node continuously must update other nodes with information. On the other hand, a distributed data architecture is vulnerable to node disconnection since the data each node possesses may not be duplicated elsewhere. The main advantage over the replicated peer-to-peer model is that the distributed peer-to-peer model requires less network traffic.

The main advantage of a peer-to-peer architecture is that you don't need a game server to connect to since every participating player already connects to every other player. Thus the operator or administrator does not need to run a dedicated server. There exist complications with this model, which makes it more suitable to be used with only a few nodes. Since the relation between number of nodes and required connections in a peer-to-peer architecture are exponential, the architecture suits only a small constellation of nodes. In equation 1, *c* is the number of connections established in a peer-to-peer network with *n* nodes (e.g. a full mesh).

$$c = \frac{n^2 - n}{2} \qquad (1)$$

Using a peer-to-peer architecture for a massive multiplayer game is not appropriate, but may suit in-game sub-services such as voice communication or textual chat communication. The use of peer-to-peer architecture for non-critical game information helps the server minimize its network load.

There exist some negative security aspects of peer-to-peer networking. The main disadvantage is that all nodes in a peer-to-peer architecture must know the addresses of all other nodes. One security issue could be that if the client software is encumbered with security holes, a person interested in exploiting these security holes could easily collect addresses of potential targets.

4.4 Access networks

Today there are many commercial possibilities for Internet access, from old-fashioned dial-up access with a traditional modem to mobile high-speed downlink packet access (HSDPA) access. The characteristics for these access links differ and many are not optimal for all forms of games. A rule of thumb is that a traditional wired packet switched network connection is to prefer over a wireless network connection, but with the evolution of mobile networks, some older access technologies such as ISDN and connections over a TV-cable network may soon see themselves outperformed by the new mobile networks. Almost all available broadband solutions are able to deliver performance sufficient for acceptable multiplayer game quality.

When it comes to mobile access networks the performance differs rather radically between the different technologies. There are two major performance updates which are relevant; one was the second generation (2G) of cellular network transition from GPRS (2,5G) to EGPRS (2,75G). The second performance change is due to the third generation (3G) transition to HSDPA (3,5G). In Table *8* (page 61) some measured performance parameters are presented from each access technology. It will be interesting as time passes to see the development towards an IP based mobile network that are truly optimized for packet switched data.

4.4.1 Delay contribution

When it comes to end-to-end delay, the delay contribution can be divided into three elements [16]:

- Delay introduced by hardware and software at the client node (i.e. delays due to the end user's equipment).
- Delay introduced by the access network.
- Delay introduced by hardware and software at the server node (i.e. delays due to the game server).

The delay generated in the end nodes (i.e. at the game server and at the player's computer) are, if the system requirements are fulfilled, of order of tenths of a millisecond and under the circumstances not decisive. Delays in the access network, the Internet, or any packet based network is a result of packet processing, queuing, signal propagation, and serialization. The final latency between two nodes is the sum of all these contributing sources. In Figure 1 illustrate a rough estimation of the delay distribution between a client and server. The actual distribution is of course strongly dependent on where the server is located, but the delay added by the client and server hardware is negligible.

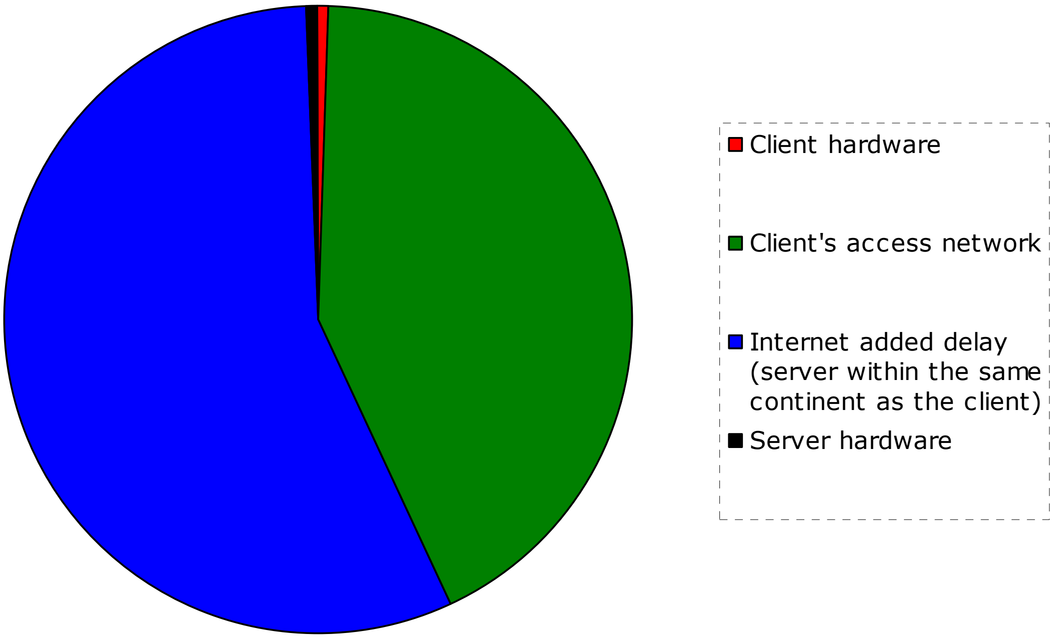

Figure 1. Rough estimation of the delay distribution between a client and server

When a packet travels from its source node to its destination node it will pass through different kinds of network equipment. One common kind of packet processing equipment is network routers. Each time a packet pass through a router a small delay occurs due to the fact that the router needs to read the packet header and decide on which interface it should be forwarded on. If a router has a high load, the packets passing through the router may have to wait for other packets before being forwarded. This kind of delay is referred to as queuing delays. Another source of packet processing delays could be Forward Error Correction (FEC) or interleaving delay [16]. FEC is a mechanism for transfer error control that adds redundant information to a message allowing the receiver to detect and correct errors. Interleaving is a method for sending bits in a more optimal, i.e. non chronological, order (helping the error correction algorithms to recover from burst errors). A disadvantage with interleaving is that there will be an additional delay, i.e.

interleaving delay, when the entire interleaved block must be received before the sent data can be retrieved. The time it takes for a device to forward the packet on the link is called serialization and queuing delay and is proportional to the length of the packet and the packets before it. Larger packets have more bits to send which results in larger delays. The last delay component is the signal propagation delay. This delay is inevitable if the signal is going to be transferred to another geographic location, since it takes time for the signal to propagate through the medium. A rule of thumb is to allocate 5 µs per km [16]. For example, the minimal signal propagation delay between Stockholm and Washington DC is 33.23 ms.

4.5 Network impairments

Online games are in general a rather demanding type of network application, with respect to the requirements in terms of latency, throughput, and packet loss. In particular real-time games are especially demanding and even small network impairments may negatively impact the game quality. More specific requirements from real time online games is studied in Chapter 5.

Since there, from a gaming perspective, is no advantage in delivering better performance than noticeable, the network's goal is to deliver sufficient online game quality for the end user. This is not always achievable and results of network impairments can easily create so called lag in the game for the end user. The term "lag" is commonly used to represent anything "funky" going on in an online game [24], i.e. anything that negatively affects the gaming experience.

4.5.1 Latency

Latency and delay are used as synonyms. There are two ways of measuring latency. The first method is to measure the delay form the client node to the server. To get reliable results, this method is somewhat tricky since you will have to synchronize the clocks of the sending and receiving nodes. There exist two common solutions when synchronizing the clocks. The first method utilize the network time protocol (NTP) synchronizing the clocks. Second method utilizes the global positioning system (GPS) and synchronizes the clocks with GPS-receivers. Both these methods suffer from uncertainty about the resolution of the time synchronization.

The second way of measuring latency, frequently used in this thesis, is to measure the round trip time (RTT). The round trip time is the time it takes a packet to travel from the client to the server and back. The advantage with this technique is that since there only is one computer involved in the time measurements, there is no need for any time synchronization. This is the most common form of latency measurements and the most popular tool to measure round trip time is "ping" available on most platforms. A disadvantage with the "ping" utilities using ICMP packets is that it is not guaranteed that the ICMP packets are routed the same path as the actual data packets. This will lead to uncertainty about the deviation between the actual round trip times for the real data packages and the round trip times for the ICMP packages that have been measured.

4.5.2 Packet loss

Packet loss occurs when a packet is sent from one node but never reaches the destination. The probability of packet loss depends on the network access medium, the load on the routers along the path, and specific "culprit" routers. In wire line networks available today packet loss rate is low. In IEEE 802.11 networks packet loss could be common and is dependent both on the current load on the access point and the current number of active clients. In wide area cellular networks retransmission algorithms are used resulting in that packet losses over the radio is converted into delay since it takes time to resend the lost packet. How packet losses are treated by the destination node depends on the protocol in use. TCP, for example, provides a reliable transmission protocol with retransmission mechanisms; while UDP is a best-effort protocol that does not know or care whether a packet reaches its destination or not.

With today's broadband accesses a few percent packet loss may occur on wireless links. The main sources of packet loss are buffer overflows in the routers (the same as the main source of loss in wire line networks), on the links, i.e. fading on wireless links, or caused by malfunctioning hardware in the end nodes.

During this thesis the method utilized for measuring packet loss is calculating the difference between the number of packets send and number of packets received divided by the number of packets sent. To keep track of the number of packets sent and received all packets were saved to a file with Tcpdump [61] for later analysis when real game traffic was involved. During the experiments when traffic generators were used to generate network load or simulate game traffic, log files were used to decide how many packets that was sent.

4.5.3 Jitter

Jitter has historically been a significant factor when designing electronic communication links. Network jitter could be described as unwanted and abrupt changes in the inter-arrival time (the time between arriving two packets) at the receiving node. The current amount of jitter in wire line networks has little impact on the game's ability to deliver game quality. In practice jitter of larger magnitude has an impact comparable with packet losses. When calculating the network jitter, saved network traces were analyzed.

4.5.4 The effect of network impairments

A central source of problems with online gaming is network impairments. The user expects to interact with other players in real-time, which is technically and theoretically impossible to deliver. The actual goal is to deliver sufficient game quality so that the user does not discover that the game does not actually update in real-time. A more detailed description of different problem caused by specific network impairments is presented below.

4.5.5 Impairments impact on end users' experience of game quality

Lag often affects the movement or action that the character is able to perform. Some common lag examples are:

- **Game freezing and game stuttering**. Game freezing is when the screen freezes for a random period of time. Sometimes game freezing is followed by *warping* (a sudden position update). Game freezes are mostly caused by long bursts of packet loss. A similar lag phenomenon to game freezing is game stuttering, but here each freeze is short in time but happens in rapid succession. The result is that the game world appears jerky.
- **Warping and rubber-banding**. Most of the more advanced games today use some kind of prediction model where the new position of an object is extrapolated from its previous position (see section 4.5). If the client loses server updates for a long period of time, then the difference between the extrapolated position and the real position may be large and there is no time to apply a smothering algorithm to the update of the player's position in the game world resulting in a (time) warp. The resulting rubber banding effect is similar to warping, but here the updates sent by the client to the server are lost, thus the client's position in the server does not correspond to the client's actual position. The result is that the client is *warping* back to the last position that the server is aware of, a so called rubber band effect.
- **Delayed responses**. For a dial-up modem user, delayed responses happen quite often. The main cause of the problem is high network latency. An example is when you perform an action with your character, but you experience a delay between your action (e.g. a mouse button click) and the moment when your action is carried out in the game world (i.e. the character shoots with his gun). The time between action and response is dependent upon the current latency between the client node and the game server node. A constant latency is preferable since the end user can adjust his game play to this latency; while latency spikes may cause irritation, especially for an experienced user.

The obvious solution to eliminate problems caused by network impairments is to minimize packet loss and guarantee stable low round trip latency. This may not be a trivial task in all access networks. Especially in mobile networks since their characteristics may vary significantly with the level of utilization by both the user and other users. Here a solution would be to implement a quality of service (QoS) mechanism that prioritizes a user's game traffic before other traffic during busy hours so that a reasonable bound to latency could be guaranteed. There exist "tricks" to hide latency for the end user (see section 4.6).

4.6 Prediction models and latency compensation

The two main objectives for a game state prediction model, when it comes to a game implementation, is to hide network impairments such as latency and packet loss from the player and to minimize the size of each game update sent between the server and the client and thereby minimize traffic. The most common prediction model is based on dead reckoning, which assumes things will continue to move as they are moving. This is described further in the next section.

4.6.1 Dead reckoning

In short dead, reckoning estimates an object's position by processing its last known position, velocity, acceleration, and travel plan [6]. The dead reckoning model was developed by the United States Department of Defense, were it was used in distributed simulation for military training, most notably in the development of the Distributed Interactive Simulation (DIS) protocol [21]. Within DIS, dead reckoning was used as a technique for latency hiding and bandwidth reduction. This technique is also very suitable for network games.

There exists several different implementations of the dead reckoning model, but the following is most probably the implementation network games utilize. The advantage of dead reckoning is that all participating nodes are capable of computing the same results. When an entity is created in the game world the game server sends out an entity state protocol data unit (PDU) to all participating nodes. This PDU contains all information about the entity that is needed to predict its future state, i.e. where it is located, velocity, acceleration, orientation, the state (health, armor, etc.) and an identifier that tells all nodes which simulation algorithm to use for this entity. When a node receives this PDU it will, if it has prior knowledge about the entity, update the entity's description with this new information, otherwise it will create a new entity and associate the new entity with the information received in the PDU. Now all clients (nodes) can display all the entities for the user, but the entities future position will be predicted based upon this description until it receives another PDU. It is here the per-entity simulation algorithm (also known as the dead reckoning algorithm) is used. Between received PDUs each node will update the location of all entities by applying the simulation algorithm specified for each entity to its previous real or simulated state. The fact that every client is simulating the same entities with the same simulation algorithms makes the outcome deterministic. New PDUs are sent whenever the difference between the real position of the entity (the owner's position of the entity) and the simulated entity exceed an agreed threshold or when the maximum time between the PDUs has been reached.

When the difference between the simulated position of an object and the actual exceed the agreed threshold value, the node that owns that object is forced to send an update PDU to the server. Notice that since every node is simulating all entities, it is a trivial task for the node that owns an object to compare the difference between its real position and its simulated position with the threshold. Once the server sends out a new update PDU to all other nodes, then every node updates their database with the position and other information about this entity. The problem now is that the actual position of the entity suddenly changes, and depending on the threshold the change of position may be perceived by the user as jerky motions. The larger the threshold (i.e. the greater the difference between the predicted position and the new position before an update) the jerkier the motion experienced by the end user (illustrated in Figure 2).

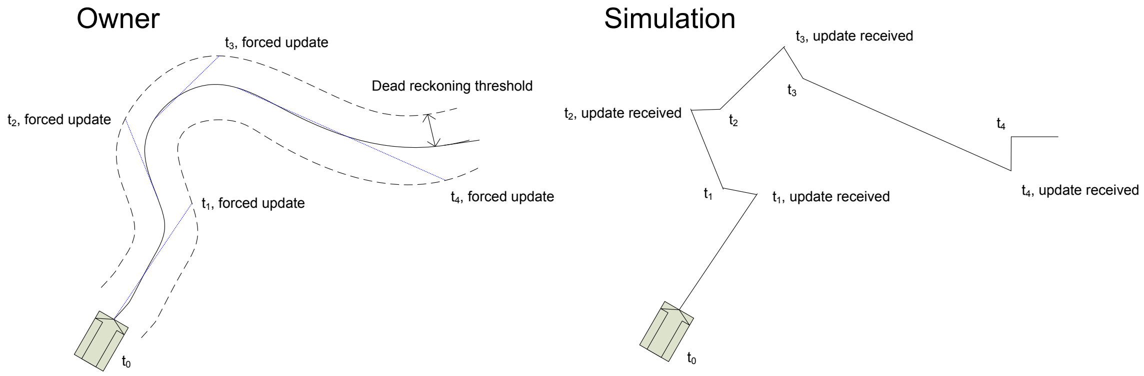

Figure 2. Dead reckoning

To get around this problem, a smoothing algorithm is used (see Figure 3). A smoothening algorithm will gradually move the entity from its position where the new PDU specifies towards the position the simulation based on the new PDU will predict after a predefined time. If this algorithm is well tuned to its application, much of the jerkiness will disappear. This model could be optimized by choosing a suitable algorithm and/or a threshold and the minimum PDU update interval. The algorithm is mostly chosen based on the type of entity. The minimum PDU update interval and the threshold determine how close the synchronization is between all nodes. Large thresholds and high minimum PDU update intervals results in poorer synchronization, but little network traffic; while low thresholds in most cases generates more frequent updates (unless the simulation algorithm are very accurate). There have been studies on this topic utilizing auto-adaptive thresholds [9].

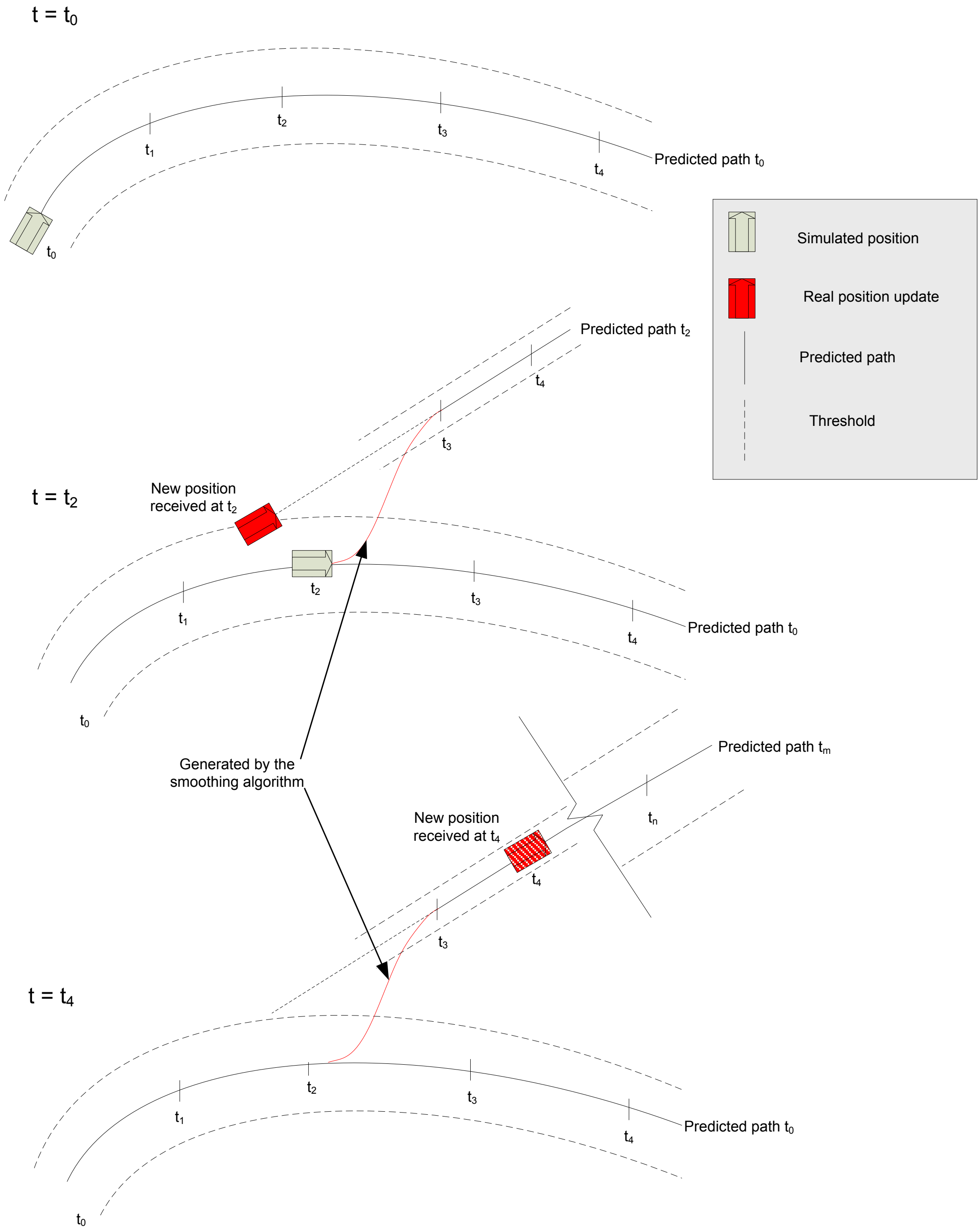

Figure 3. Dead reckoning with smoothing algorithm

4.6.1.1 Benefits

The clear benefits of dead reckoning are fewer game updates to be by from the server since game state updates (i.e. PDUs) only are sent when actually needed. A second benefit is that problems with latency to some degree are hidden for the user.

4.6.1.2 Disadvantages

The continuous simulation of all entities requires CPU resources and on low performance systems this may be of concern. Another situation were dead reckoning performs poorly is when the routes of entities are very hard to predict and hence hard to simulate. This will cause a high rate of server updates and can cost more (with respect to both of extra CPU-cycles (for the simulation/extrapolation) and network traffic) than gained from prediction. However, for large scenarios DIS has shown that dead reckoning offers significant advantages over non-dead reckoning implementations [21]. Another concern with dead reckoning is that it is vulnerable to cheating. A player could, theoretically, pretend to have greater latency than he actually has in order to see actions of other players before sending his own responses, which gives the player a way to see into the virtual future. One approach to ensuring fair gaming is to introduce a delay in the message processing in the server so that every client must send their updates before the server processes the information. The clients that have not sent their update before the delay limit will be updated with help of dead reckoning [1]. The results of this approach do not seem to have been studied closely.

4.6.1.3 Additional notes

It is possible to apply the dead reckoning to more properties of an object. For example the orientation of a tank turret could also be predicted. Game developers have lately been working with more and more advanced game physic models, and today it is even possible to purchase physic processing units [60] to help the graphical processing unit in with advanced physical models.

In the gaming industry dead reckoning has successfully been applied to first person shooters, sports games, flight simulators, and massively multiplayer games [1].

4.6.2 Predictive contracts

To help the simulation algorithms to predict non-obvious paths the entities could be associated with a "travel plan", a so called *predictive contract*. In dead reckoning all players presume that the object is doing exactly what it was doing when last updated. Predictive contracts enable entities to look much further ahead. An example would be a car positioned on a road driving in a known direction with a known velocity and acceleration. In this example the predictive contract could state "follow the right side of the road towards the waypoint X" since it is most likely that the driver of the car (i.e. a user) will both follow the road and drive on the right side of the road (in countries where cars drive on the right). The user that owns the car now only has to update his current position, velocity, acceleration, and the predictive contract he will be using. The advantage is that the player need not send any updates unless the user deviates from the predictive contract.

An even more refined way to use predictive contracts is if the node that owns an object learns the user's behavior, i.e. the user's driving pattern. Based on this information a new predictive contract is automatically created and included in an update to the rest of the nodes.

One requirement for using predictive contracts is that every participating node has the same definition of the predictive contract in use. Predictive contracts are seen as an extension to dead reckoning [21].

4.6.3 Visual tricks to hide latency

To hide latency from being perceived by the user, game developers have implemented many different kinds of visual tricks.

One effective way to hide latency is to give an immediate local response to the user. One example could be that the user's weapon fire immediately when the game receives the user's input, but the eventual damage caused by the shot must await server's response. The drawback is that the action may be seen as illegal by the server; hence it is subject to being overridden in next server update.

Yet another trick is the use of an animation of variable length when an action has to be confirmed by the server. The animation is used to hide the delay until the action can be carried out. An example is when a tank has to stabilize it's gun turret before it can fire.

4.7 Data compression

To reduce the bandwidth utilization, a common technique is to use data compression. Since waiting until a large number of packets has arrived until uncompressing, a per-packet compression method is used. Often the use of more advanced compression methods costs more in processing than the gain due to compression. The result is that processing-friendly methods are commonly used. An example of such algorithm is Huffman coding. This and other techniques are examined below.

4.7.1 Delta compression

Delta compression assumes that much of the data describing the full game state is the same over several updates; hence delta compression can exploit this redundancy. Instead of sending the full game state every update, only entities that have changed since last update are sent; additionally within a specified period of time, the full game state is updated. One popular delta compression scheme is the protocol independent compression algorithm (PICA) [54]. PICA is protocol independent since it performs delta compression at a bit level and removes redundant bits compared to the previous transmitted update.

The main disadvantage of PICA and other delta compression schemes is that they require reliable message transportation since a game state update requires that the previous updates have all reached the receiver. Without any retransmission algorithm the game state would be incomplete until next full game state is transmitted. Since game traffic often utilizes UDP a retransmission protocol needs to be developed to be used above the UDP protocol. Since updates are frequently sent from both the server and the client, the implementation is generally that ACKs and NACKs are usually piggybacking in these packets. A common approach to minimize delays is frequently sending duplicates of packets until the sender receives an ACK for that packet.

4.8 Game traffic compression ratio

$$Compression \quad Ratio = \frac{Compressed \quad Size}{Original \quad Size}$$

The compression ratio of transferred data (usually) gives a hint of how much the data traffic could be optimized. It is common that games use a CPU-cycle friendly compression algorithm to perform per-packet compression. Since online gaming is a real-time service, compression over multiple packets is usually not possible, which together with the delta compression scheme presented above well explains the poor compression results presented in Table *3*. In general, the lower the compression ratio the greater the advantage of performing compression of the data *before* it is sent. As could be seen, the measured compression ratio for all games was less then 0.5 which implies that the transferred data volume could have been compressed to less than half the size, if traditional compression algorithms would have been applicable on gaming traffic.

Table 3. Game traffic compression statistics

Game	Original dump file size (headers + data) (kB)	Total data size (kB)	PKZIP compressed file size (kB)	PKZIP compression ratio	Bzip2 compressed file size (kB)	Bzip2 compression ratio
Battlefield 2 (dedicated server)	672388	617356	226864	0,337	207512	0,309
Battlefield 2 (client)	17408	15893	4188	0,241	3584	0,206
DOOM 3 (client)	5312	4645	2172	0,409	2100	0,395
Quake 2 (client)	2748	2345	1008	0,367	988	0,360
Quake 4 (client)	592	514	288	0,486	276	0,466
Soldat (client)	620	505	252	0,406	236	0,381

4.9 Online cheating

The goal for online cheaters is to gain an unfair advantage against the other players. Online game cheaters are seen, by most gamers, as similar to how doping is seen on by athletics (i.e., it is a bad thing if someone else does it and is not caught). Unfortunately cheating exists in all online multiplayer games in some way. The exact definition of cheating differs from game to game, but could range from modifying player skins to decoding game traffic to learn information about one's opponents.

4.9.1 Types of online cheats

In online games there exists many ways for a user to improve his/hers performance in an online game. Most of them are considered cheating, but some are not. The boundary is in some cases subtle. The most common cheats are to modify game client configuration, using external cheating software, or unfair user behavior.

The evaluation of an action differs between games as to what kind of client modification is considered as cheating. In most games the user is free to configure game settings like input bindings, graphics brightness level, etc. to please the user and suit his or her system. Most of these are generally not considered cheating. In first person shooter games it is common to modify the player skins and textures. It is popular to modify skins to be easier to discover in dark areas of the game levels. This is on border of what most players tolerate since it would give them an advantage against players playing with the original skins.

The use of external software to cheat in online games is perhaps the most effective and most disliked form of cheating. One approach is to use an application that listens to game updates from the server to modify the game updates sent from the client. The application could reveal the positions of all players that were included in the server game update to the user, even players that are out of sight of the cheater. Not surprisingly first person shooter “aimbots” have caused serious problems. The goal with an aimbot is help the cheater reach a higher hit accuracy. The aimbot could automatically shoot on coming enemies or modify the directions of the cheating user’s bullets to steer them towards one’s opponents.

Unfair user behavior is not cheating in the same manner as described in the previous two examples. The main reason why users may show impudent behavior is that he or she feels safe behind his or her nick name, believing that it provides some degree of anonymity. An example of such behavior could be to disconnect from the game server if the opponent has the lead and if the game server keeps track of the number of wins and losses for each user, the cheater will not get his or her loss counted and the potential winner will not get his or her win counted.

In general the more of the code that is run on the server the less probably the cheating will be successful (i.e. the less control the client has, the fewer possible ways to cheat). To reduce the number of cheaters within a game, so called anti cheat technologies are used.

4.9.2 Online cheating; detection and prevention

Online cheating is a problem for game developers since it is reduces the game title’s attractiveness among honest gamers and thereby the willingness of possible consumers to buy the game. Due to this game developers and third party software developers have created technologies to prevent users from cheating. One of these technologies is PunkBuster from Even Balance [35]. PunkBuster exists of additional game code that is being run both at the server and at the client. These kinds of technologies in general, but PunkBuster in specific, offer functionality to:

- Scanning the client's memory searching for known cheats.
- Provide an automatic update facility.
- Individual clients can be checked for exploits of the game engine [35].

If the cheat preventing software finds a cheater, the most common consequence is that every players on the game server is informed that the gamer was found with some sort of cheat, whereupon the user gets kicked and banned from the server. The ban is often limited in time, but since the game servers are connected through the cheat preventing software it is often possible to blacklist a player to prevent him or her from playing the game on any server that have the same cheat preventing software enabled.

4.9.3 Cheat prevention's impact on game traffic

Cheat prevention creates additional uplink and downlink traffic since tests are performed continuously (in a somewhat random manner). The actual traffic can be test results that are compared by the server to the expected results or screen dumps to verify that the user does not use any cheats that gives the user visually advantages. In general, anti cheat software does not generate much traffic; the most obvious disadvantage is that running these tests requires CPU cycles. Unfortunately CPU cycles are also coveted by the game software itself. If the CPU utilization at the client terminal is high and the requirements from the game high, the additional requirements due to the cheat prevention software may result in both reduced client game performance and a poor perception of game quality. With reduced game performance, most games will reduce frequency of uplink client game updates. An obvious distinction between game traffic and cheating prevention/detection is that these software can send its traffic at a much lower priority than the game traffic since it is unlikely to be time critical.

4.10 Scalability

Scalability in multiplayer games concerns the ability of the game to dynamically adapt to the number of players. In most games the size of the virtual game world is limited both in its size and the number of players that are allowed to be simultaneously connected. An example of a game that adjusts itself is the first person shooter game "Battlefield 2" where the game server adjusts the game world according to the number of players connected [59]. How the number of simultaneous players influence the game traffic is treated in section 5.7.

Chapter 5
Game traffic in wired packet switched networks

In this chapter we will take a closer look at the traffic that personal computer based games generates while communicating, the connection between network impairments and degradation of game quality, and what requirements are placed on the network to guarantee acceptable game quality for the end user. The games used in this and next chapter has been carefully been chosen since they belong the group of the FPS games that places the highest demands and that they all are representative of the games studied in this thesis. All these games are optimized for wired networks. The traffic traces were collected in a controlled environment. Even if there are indications that the results might be applicable to some game titles in general, there is no statistical significance in the analysis of this data to support such a generalization.

This chapter consists of three main parts. The first part describes the laboratory hardware used when collecting traffic. Second part discusses game traffic and traffic parameters for both the server discovery traffic and the regular game traffic but also how the traffic is influenced by a growing number of users, e.g. scalability. The third part treats the network performance parameters and the connection between network performance impairments and game quality degradation.

5.1 Traffic collection and analysis

To analyze multiplayer gaming game traffic traces are need. This report is based traffic generated in a laboratory test-bed where, in most cases, the control of all environment parameters was high. The goal of collecting own traffic was to utilize a very controlled environment where we controlled all relevant parameters. This test-bed is described in the next section.

5.1.1 Test bed setup

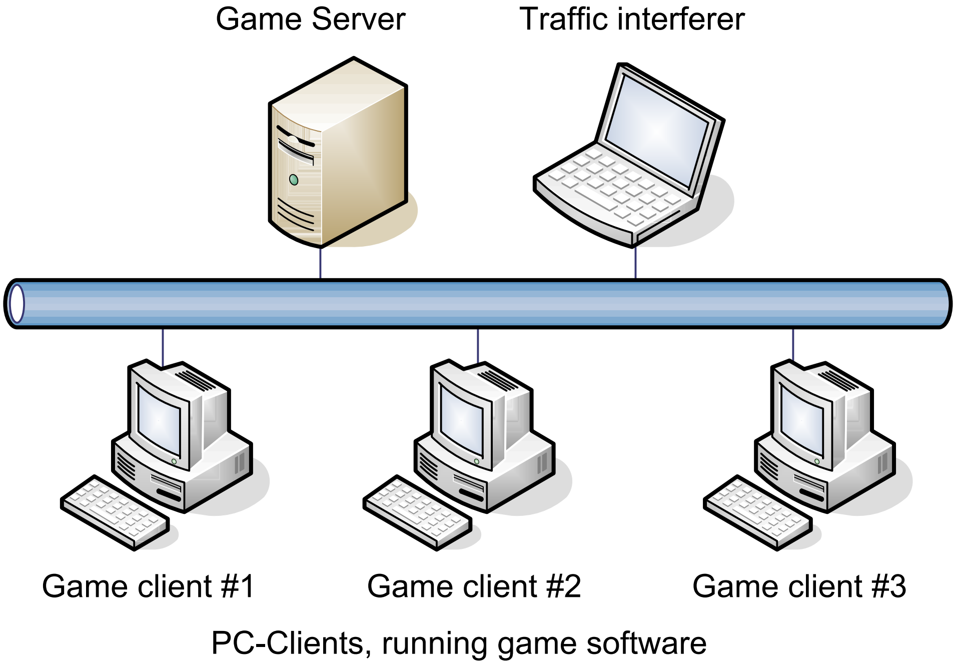

Figure 4. The test-bed

The main goal of this laboratory test-bed was to generate game traffic in a controlled environment under known conditions. The requirements on the game clients were to be able to successfully run the game software without any visible problems. This requirement was set since game quality impairments caused by malfunctioning or inadequate hardware in many cases are similar to game quality impairments caused by network problems (in which our interest rests). By using a state of art collection of clients and servers, we believed that any degradation in game quality could only be due to network impairments. The main requirements on the server were to be able to run the dedicated game server software without any problems and be able to provide the gaming service to all connected clients and in parallel running Tcpdump to save traffic traces for later analysis. A laptop was used both to disturb the traffic on the network and to simulate different types of access networks based on known characteristics. The testbed consisted of three game clients, one game server, and one laptop (see Figure 4). Some initial tests showed that the equipment specified in Table 4 fulfilled all requirements that were put on the equipment.

Table 4. Terminal specification

Equipment specification	**Game clients**	**Game server**	**Traffic interferer**
CPU	Intel P4 650 3.4 GHz 800 MHz 2 MB cache	Intel P4 3.2 GHz 800 MHz 2 MB cache	Intel Pentium M 1.8 GHz 2 MB cache
Video card	ATI Radeon X1800GTO 256MB PCI-E		ATI Radeon Mobility 9600
Memory	1 GB DDR2 PC4200 533 MHz	1 GB DDR 400 MHz	512 MB DDR 333 MHz
Hard drive	40 GB 7200 RPM	40 GB 7200 RPM	30 GB 4200 RPM
Network	1 Gb/s	1 Gb/s	1 Gb/s
Operating system	1x Windows XP Home edition 2x Windows XP Professional edition	Linux, kernel 2.6.15	Linux, kernel 2.6.15
Viewing device	17” low latency LCD		14” LCD

In addition to the computer components mentioned a fast Ethernet switch and cat 5 Ethernet cables was used.

5.2 TCP, UDP, and gaming

In brief, the main difference between TCP and UDP is that TCP is a connection oriented reliable in order delivery protocol, while UDP is a connection less unreliable datagram protocol. If all functionality that TCP provides came at no cost, it would be the choice. Unfortunately what TCP guarantees costs both in bandwidth (due to handshaking, packet acknowledgements) and delay (due to retransmissions, etc.). The drawback with UDP is that the packets you send may or may not arrive the destination and you will not know if the datagram has arrived or not, unless you add your own acknowledgements. Even if the functionality TCP offer fits gaming rather well, in general UDP is more suitable for real time game traffic due to the overhead of TCP. Today almost all online games rely on UDP traffic. Many games also uses delta compression, i.e. sending game state updates that tell only what has changed since the last update, which results in packet being more important – since the state can only be kept consistent if all packet reach their destination. To ensure this, a retransmission algorithm is used which is similar to TCP. An example of the use of TCP for game communication is the primitive network implementation in the classic FPS game Quake.

5.3 Traffic types

There exist three types of game traffic that are generated by a game client. The main category is regular game traffic whose purpose is to keep the game world state updated between the server and the client. It is this category that places the main performance requirements on the network. The second category is pre-game or server discovery traffic. Included in this category is all traffic that is generated prior a game session while the user is looking for a suitable game server to play on.

In first person shooter games it is common that the game developer offers a game server indexing service, a so called “master server”. Each public game server then registers itself with an available indexing service. When a player wants to find a server to play on, he or she asks the indexing service for a list of available game servers, which can contain thousands of game servers. The next step for the client is to iterate through the list and query each server about its current status. The game server status contains information that forms a basis for the user’s choice of game server to play on. Such information could include the current map that is being played (i.e., a region in a virtual space where the game is being played), number of connected players, number of total client slots available, and what rules the server apply.

The third category of game traffic is background traffic. This traffic concerns the game session being played, but does not directly affects the current game state. The information transfer could include software patches that are distributed for the game client binaries or anti cheat data. Anti-cheat information could range from a checksum of the client game binaries or an actual screen shot of how the game world is rendered for the end user.

Within the first type of traffic (in-game traffic) there are two sides generating traffic: the client and the game server.

5.4 Server discovery traffic

As earlier mentioned there are master servers that provide users with information about available online game servers. To get updated status information for all these servers, each server is individually queried (probed) by the client. Depending on how traffic intensive a game is and on the size of each individual probe, the sum of all game probe traffic could be several percent of the overall game traffic volume. An earlier study on Quake 3 shows that 6-7% of the overall game traffic for a popular Quake 3 server is probe traffic and up to 35% for a less popular server [38].

To get a better understanding of the game probe traffic characteristics a Battlefield 2 server was run for 24 hours while all traffic to and from the server was captured. This game server could be classified as unpopular since it had only run for 24 hours, thus only a couple of real game session were played during this time and this server was competing with roughly 6800 other online Battlefield 2 game servers. The incoming game server probes per second over the day are shown in Figure 5.

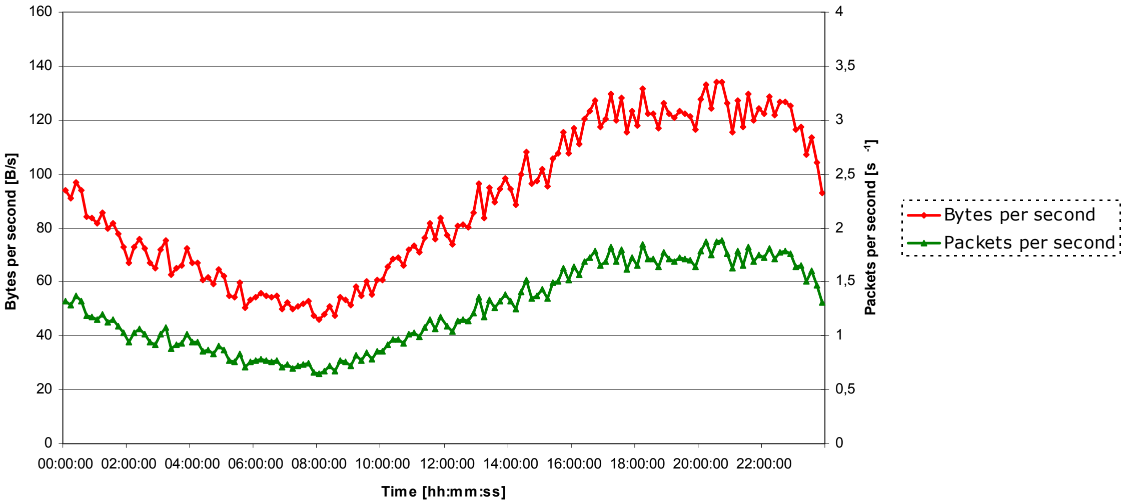

Figure 5. Server discovery traffic from a 24 hours Battlefield 2 server trace

As could be observed in Figure 5 the server discovery probe rate potentially follows a daily game usage distribution; however, since the data was only collected for a server which was connected for 24 hours there are no statistical basis for this claim. This sidereal behavior could be due to some sort of location based update rule which gives priority to game servers that are geographically close to the gamer.

Table 5. Game client server probes (per server)

Traffic direction	Number of packets	Packet size	Protocol
Battlefield 2			
Uplink	1	72 bytes	UDP
Downlink	1	150-200 bytes	UDP
Quake 4			
Uplink	1	56 bytes	UDP
Downlink	1-2	900-1600 bytes (average 1125bytes)	UDP

Let us assume that the Battlefield 2 master server returns 6000 game servers to a client query. We assume that the information shown in Table 5 that was received from a Battlefield 2 server traffic trace is valid, thus the average downlink packet size is 165 bytes. To iterate through the server list would generate 422 kB uplink traffic and 966 kB downlink traffic. For the client this traffic is only generated once every time the game is looking for a new server to play on, but the server will receive and send discovery traffic every time a client updates his server list. This traffic is generated independent of factors such as geographic location, popularity, and server bandwidth.

One big difference between server probe traffic and general game traffic is that server probe traffic can originate from anywhere in the world, while users tends to select servers that are close to them geographically. Some newer game titles, such as Battlefield 2, may have adopted this strategy while older titles may show the ineffective behavior to probe servers that most likely won’t be a game server that is finally chosen.

5.5 Game traffic characteristics

The general game traffic characteristics differ in many aspects from regular Internet traffic characteristics generated by applications like web-browsers, mail clients, instant messengers, or file-sharing programs. The general nature of game traffic is that it consists of small packets that are sent in regular bursts. In most games, the data that is sent in an update is less than 1000 bytes and fits therefore generally into a single IP packet (depending on MTU). This traffic can be divided into two categories: server-client traffic, *downlink*, and client-server traffic, *uplink*.

Uplink traffic mostly consists of movement instructions, positions, field of view, and game state information. The downlink traffic contains updates to the game world. Details of the game world update are limited to a fixed or variable radius around the player for two reasons. The first (1) is to reduce the bandwidth needs (as it is unnecessary to update the game world that is out of the player's field of view) and second (2) to minimize the possibility of cheating and thereby gaining advantage over other players. The result is that when the server sends out game state updates, each client will get his own "unique" update.

In general there are more packets sent per second in the uplink direction while the packets sent in downlink direction are larger. When it comes to traffic volumes, due to the faster rate uplink (due to the aggregation of traffic from many players), the uplink traffic volume may be more than double the downlink traffic volume.

During a game session the game traffic is rather easy to predict. The largest divergence happens when special recurrent events occurs in the game. Such events could be end of a round, end of the game, a map change, or when a player's character dies. During these events the traffic volume could either raise due to increased game state information needing to be updated or the transferred traffic volume may drop due to decreasing needs to have an up-to-date game world.

In the following sections packet sizes, inter-arrival times, and packet rates is discussed. During the collection of the data that these sections are based upon, the setup shown in Figure 4 was used. In all cases where the number of players on the server is not mentioned, two simultaneous players were connected to the server.

5.5.1 Packet size

To keep the game world updated both the server and the client need to receive and send packets. The information content of each packet depends both on which side is sending the packet and on the current game state.

5.5.1.1 Uplink packet size

Each client controls a limited number of objects. Client controllable objects include the character itself, equipment, and vehicles. In its updates to the game server the client attaches information about all the relevant objects he controls. In most games each player possesses a near constant number of objects whose state must be synchronized with the game server. This results in a very homogenous packet size distribution (within each game session). In Figure 6 the average client packet size for each game is shown and in Figure 7 the packet size distribution for Quake 4 and Battlefield 2. In addition to game state information packet acknowledgement receipts are sent piggy-backed on the client packets.

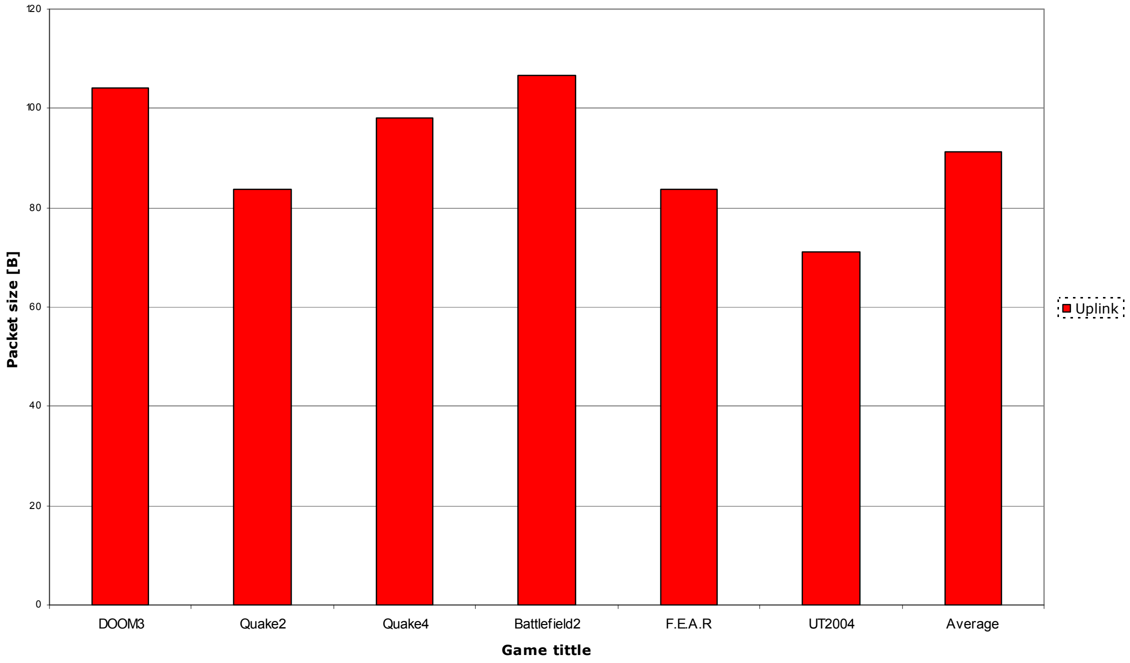

Figure 6. Average packet size per game

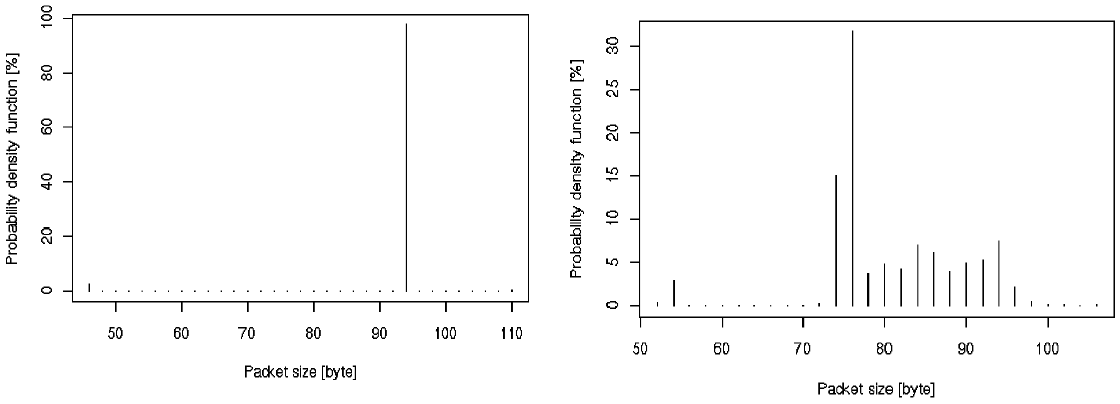

Figure 7. Battlefield 2 (a) and Quake 4 (b) uplink packet size distribution

5.5.1.2 Downlink packet size

The server traffic packet size is not as homogenous as the client's traffic. The server needs to update a client with all relevant information that has changed since last update. Most games implements some sort of delta compression which limits the amount of data that needs to be transferred, but on a longer time span, all information has to be updated. The more action and motion the client is surrounded by, the larger will each update be. In general downlink game packets have an average size of 150-200 bytes, but packages of up to 800 bytes occur. In Figure 8 the average downlink package size per game is shown.

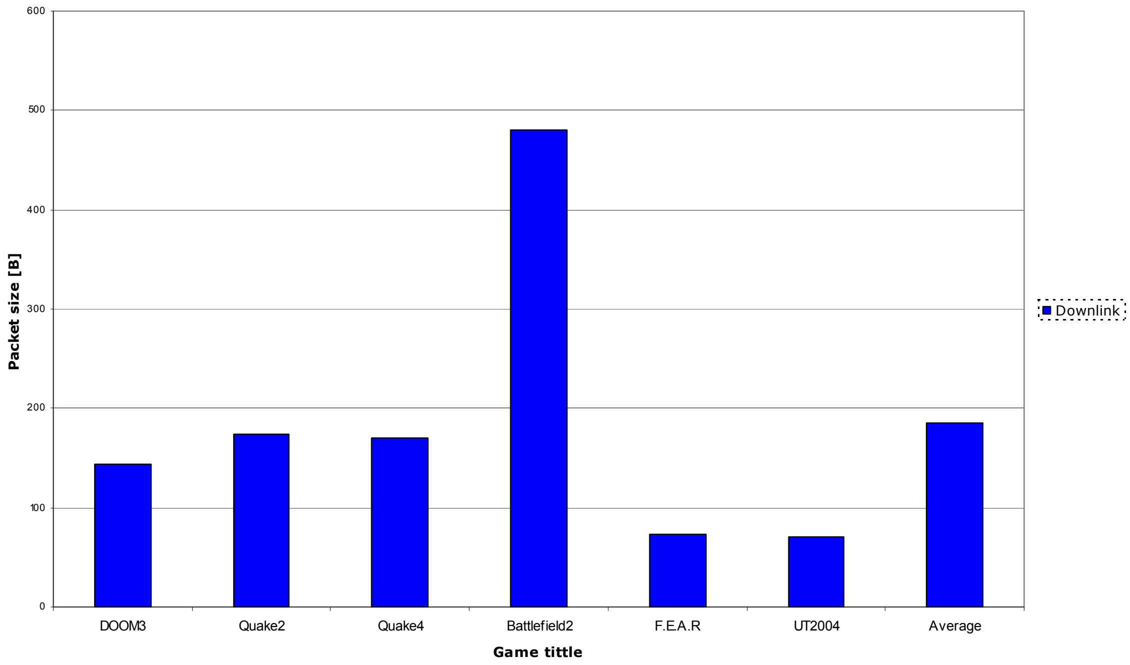

Figure 8. Average downlink packet size per game

In Figure 9 and 10 both the packet size distribution and cumulative packet size distribution from an eight player Quake 4 and a Battlefield2 trace are illustrated.

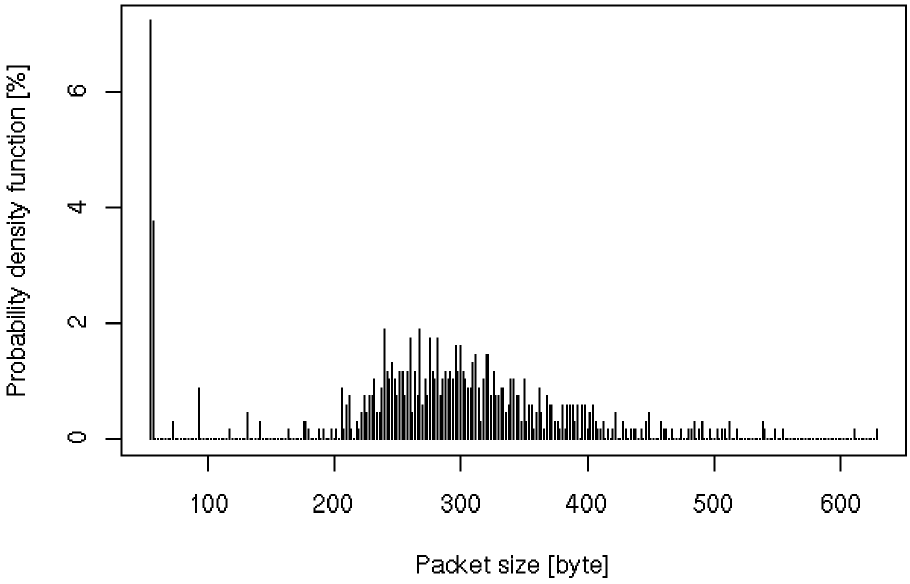

Figure 9. Quake 4 downlink packet size distribution

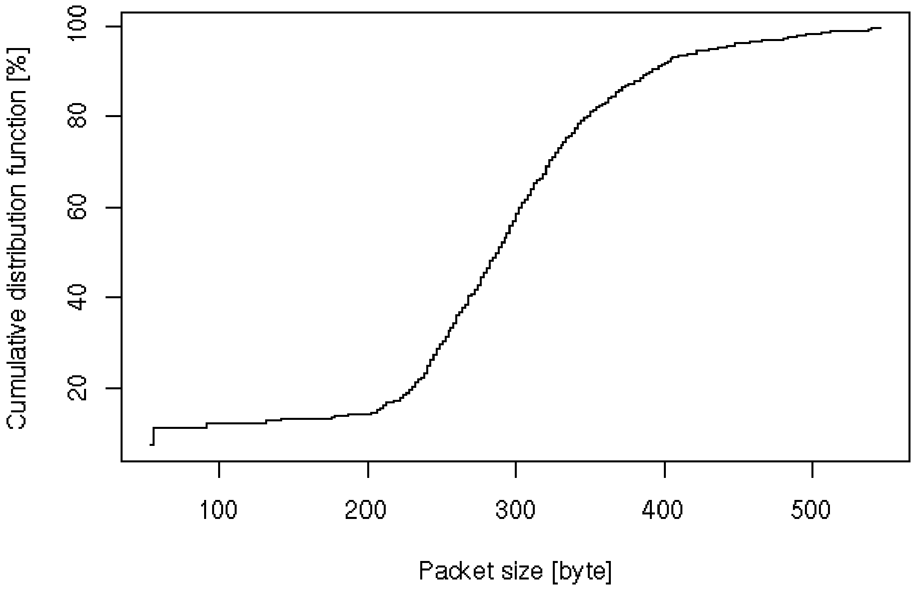

Figure 10. Quake 4 downlink cumulative packet size distribution

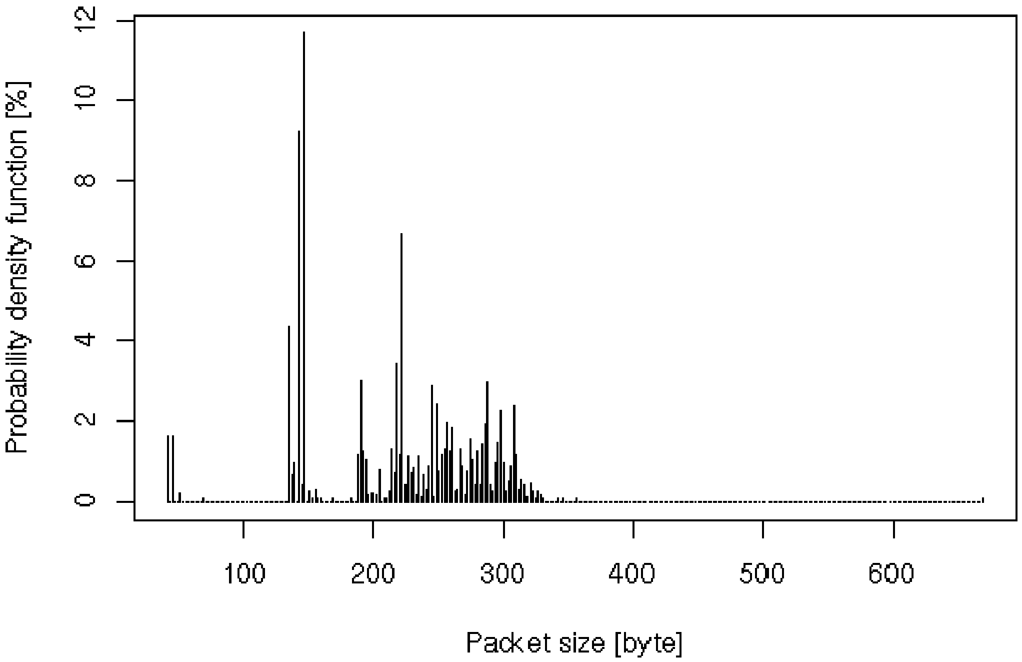

Figure 11. Battlefield 2 downlink packet size distribution

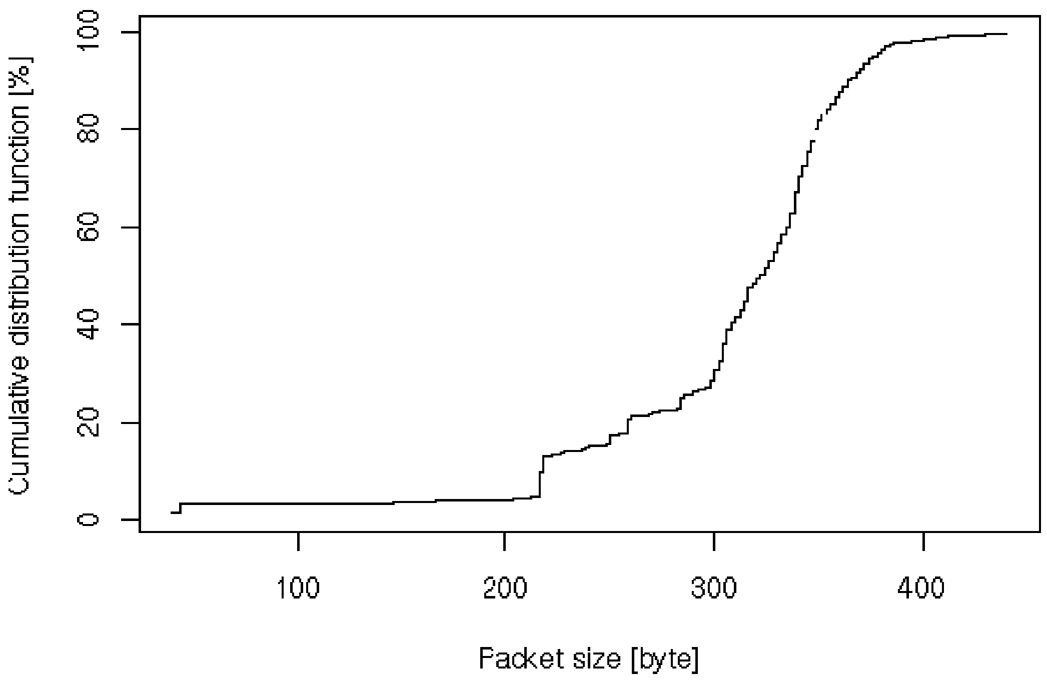

Figure 12. Battlefield 2 downlink cumulative packet size distribution

5.5.2 Packet rate

Online real-time multiplayer games endeavor to maintain a synchronized game world. At the same time as the client tries to keep its game objects' positions up-to-date at the server, the server tries to maintain as synchronized game world as possible by sending game state updates to all clients. In general, within a short period of time, the packet rate is very regular.

5.5.2.1 Uplink packet rate

The uplink packet rate is tight coupled to the terminal's ability to update the game world for the end user. Most games have some kind of limitation upon the maximum number of packets that are sent per second. If the terminal has enough performance to run the game in a near optimal way, the packet sending rate will be nearly constant during the game session. Figure 13 shows the typical uplink packet rates for the tested games.

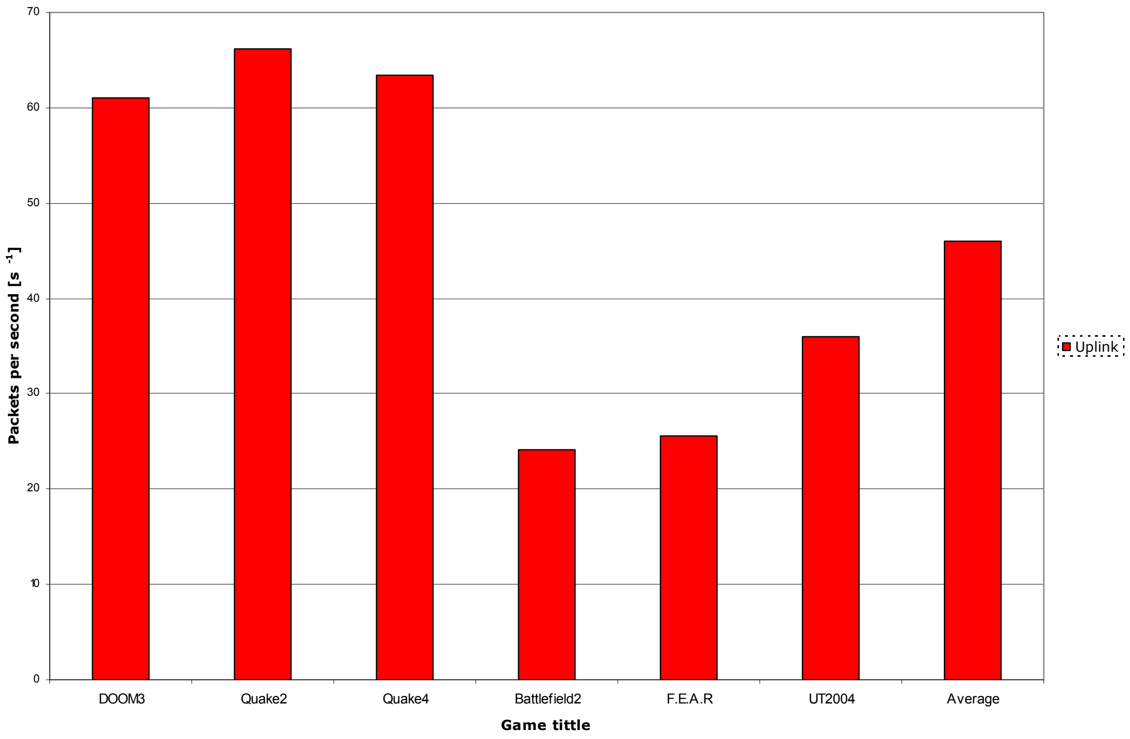

Figure 13. Average uplink packet rate per game

5.5.2.2 Downlink packet rate

The downlink packet rate is more evenly distributed between the different tested games. The mean packet sending rate value is about 20 packets per second; which occurs in all tested titles except UT2004 (Figure 14).

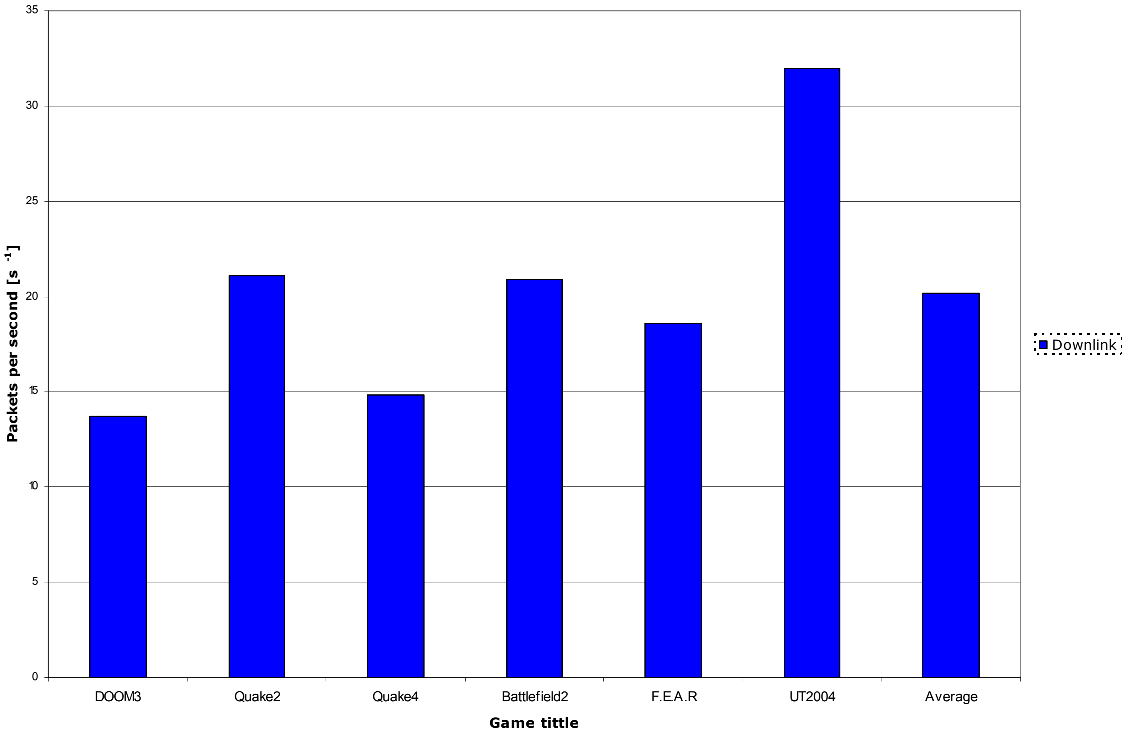

Figure 14. Average downlink packet rate per game

5.5.3 Inter-arrival time

Just as there were difference in both packet rates and packet sizes, there are obvious differences between uplink and downlink packet inter-arrival times. These differences are treated below.

5.5.3.1 Uplink packet inter-arrival time

In conformity with the uplink packet rate, the uplink packet arrival time depends on client hardware and its ability to update the game world for the user. The difference in the uplink packet inter-arrival time distribution could sometimes be striking as shown below.

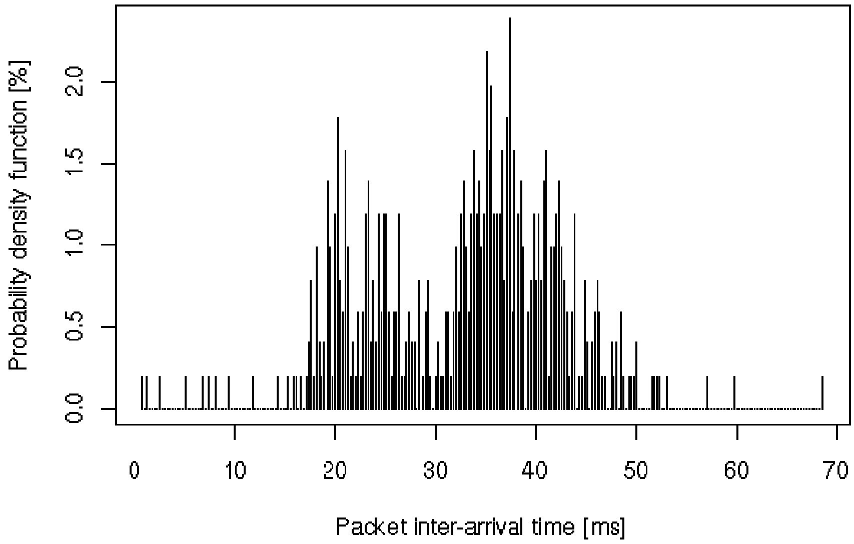

Figure 15. Battlefield 2 uplink packet inter-arrival time distribution

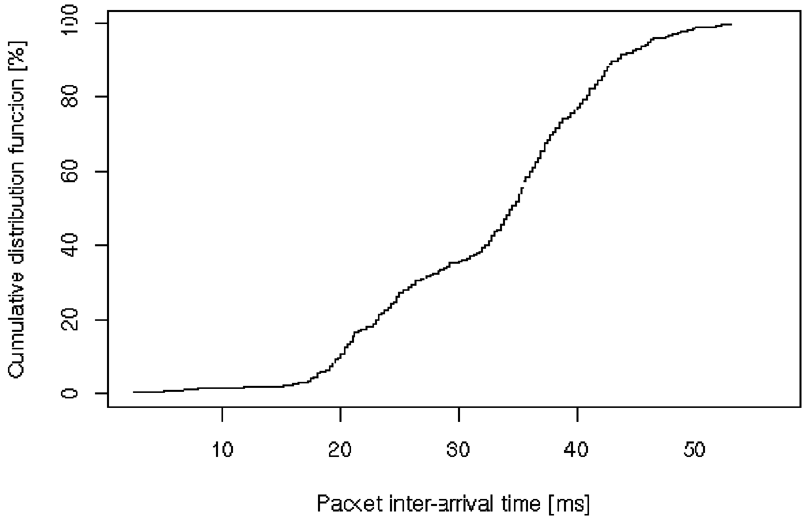

Figure 16. Battlefield 2 cumulative uplink inter-arrival time distribution

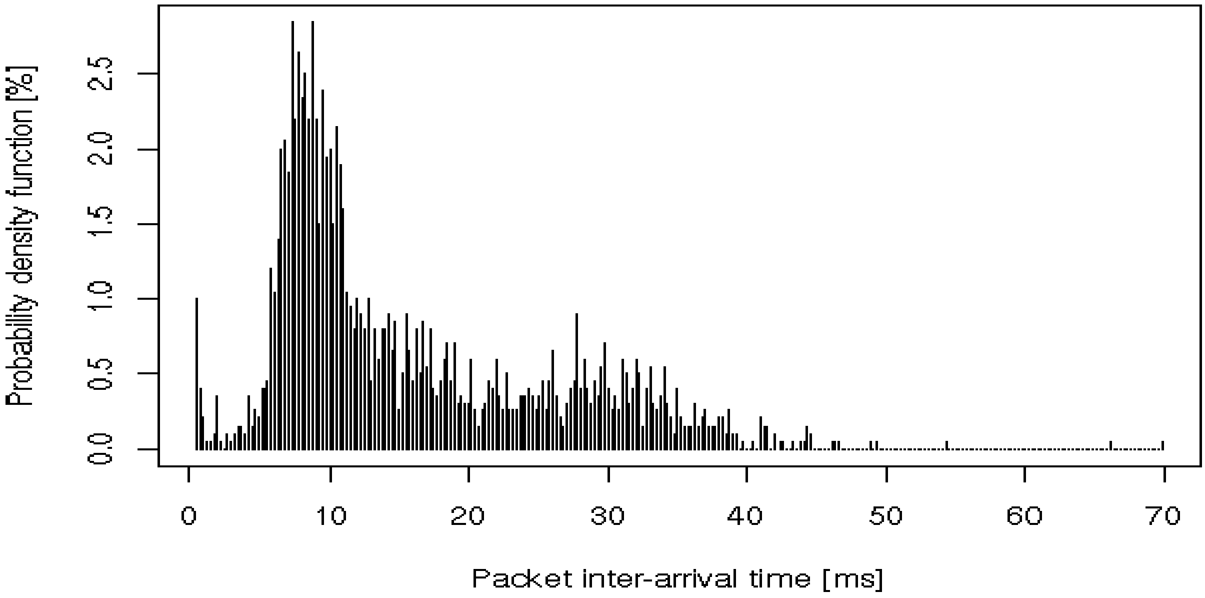

Figure 17. Quake 4 uplink packet inter-arrival time distribution

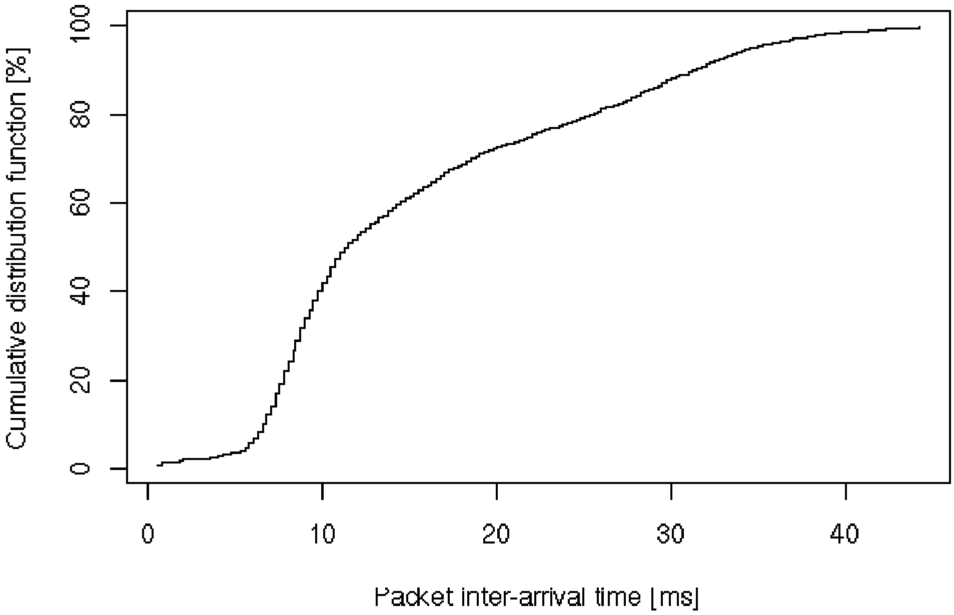

Figure 18. Quake 4 cumulative uplink packet inter-arrival time distribution

5.5.3.2 Downlink packet inter-arrival time

The downlink packet inter-arrival time in all tested games was regular. Given the packet rate, most games have an average inter-arrival time around 50ms; with Quake 4 showing a somewhat deviant behavior which is illustrated in the figures below.

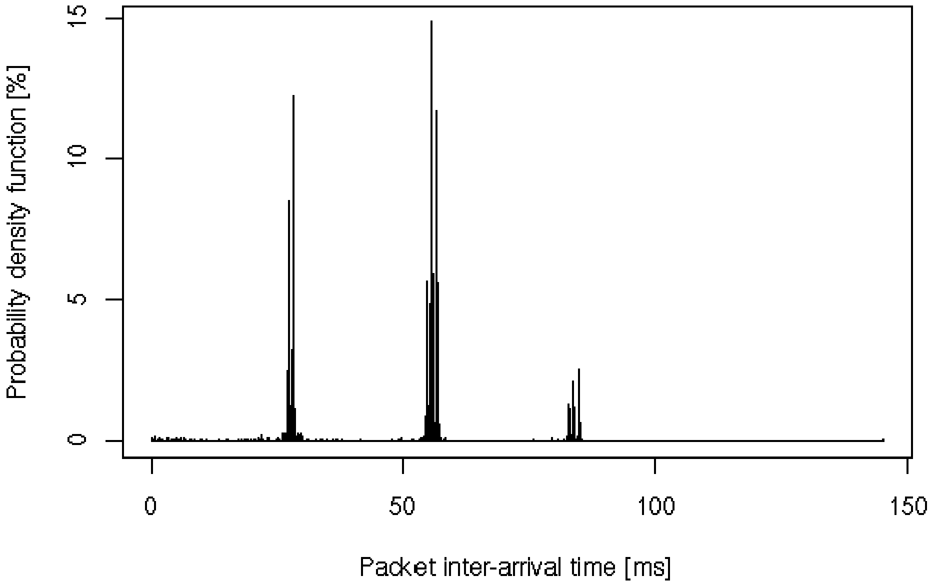

Figure 19. Battlefield 2 downlink packet inter-arrival time distribution

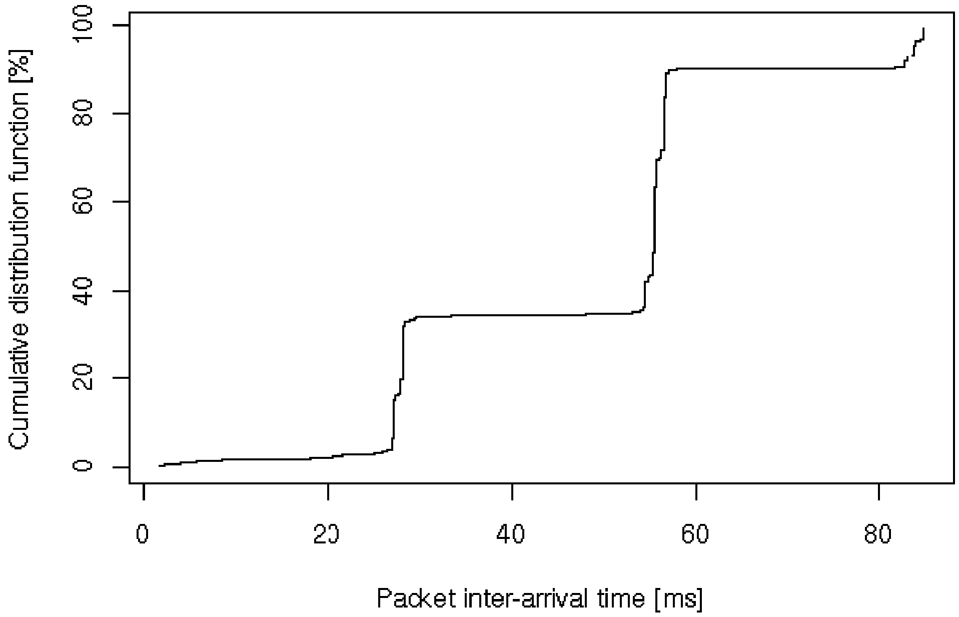

Figure 20. Battlefield 2 cumulative downlink packet inter-arrival time distribution

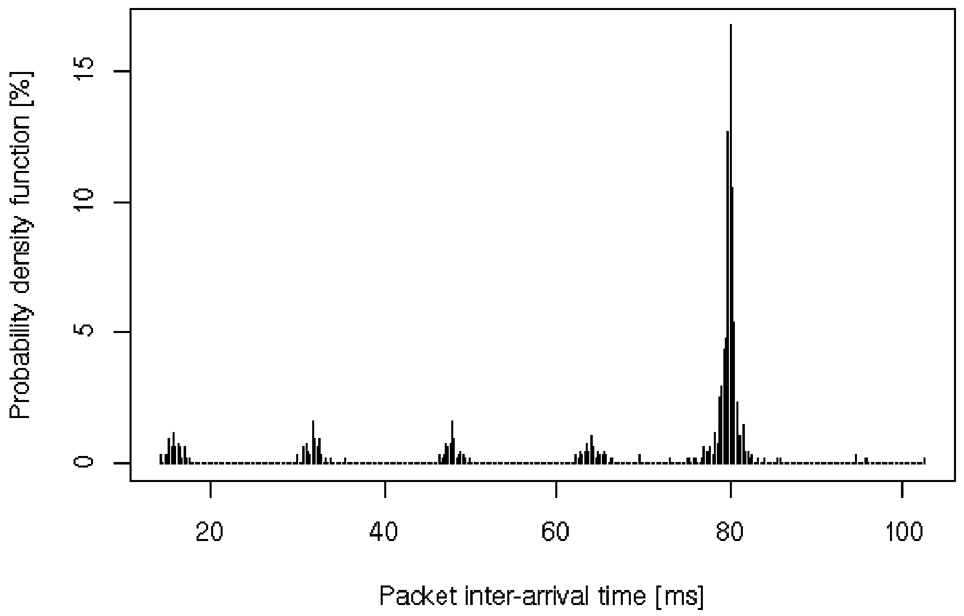

Figure 21. Quake 4 downlink packet inter-arrival time distribution

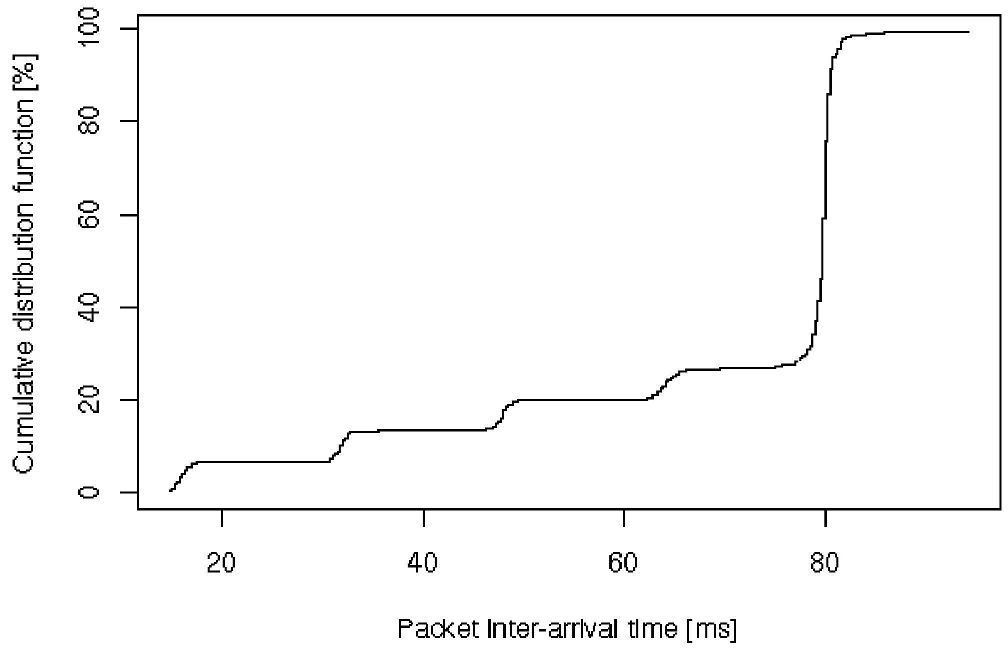

Figure 22. Quake 4 cumulative downlink packet inter-arrival time distribution

5.6 Game traffic simulation

Simulation of traffic in general and game traffic in particular could, for example, be achieved with the network simulator ns2 [41]. With a game's traffic characteristics realistic game traffic could be generated. This has been studied in several papers before [31, 55] and will therefore not be treated in greater depth in this thesis.

5.7 Game parameters' impact on game traffic

Variances in game traffic could most often be derived from the state of the game world. Three parameters that have high impact on the packet size is the game world size, number of players, and the game entity density.

To save network resources game developers want to send as small game update as possible. As mentioned in chapter 4 each packet is a game state delta compression (i.e. the changes in the game state since last update) and each packet could even be compressed once more with a per packet Huffman compression [10]. In most applications it is unnecessary to give the user a complete updated game state. A common solution is to have an up to date game world bubble around each user which is adjusted based upon the current position and velocity of the user (the game world updates can cover a larger surface when the user is flying an aircraft and cover a smaller subset of the game world when the user is in a small room). Each delta game state update is proportional to the number of entities that need to be updated.

5.7.1 Number of players

The traffic to and from the server varies with the number of players currently active on the game server. Figure 23 and Figure 24 shows the results of a test where we investigated the impact of the number of players on the packet size. No signs were found in the games we tested that the number of players has any influence on the upstream packet size.

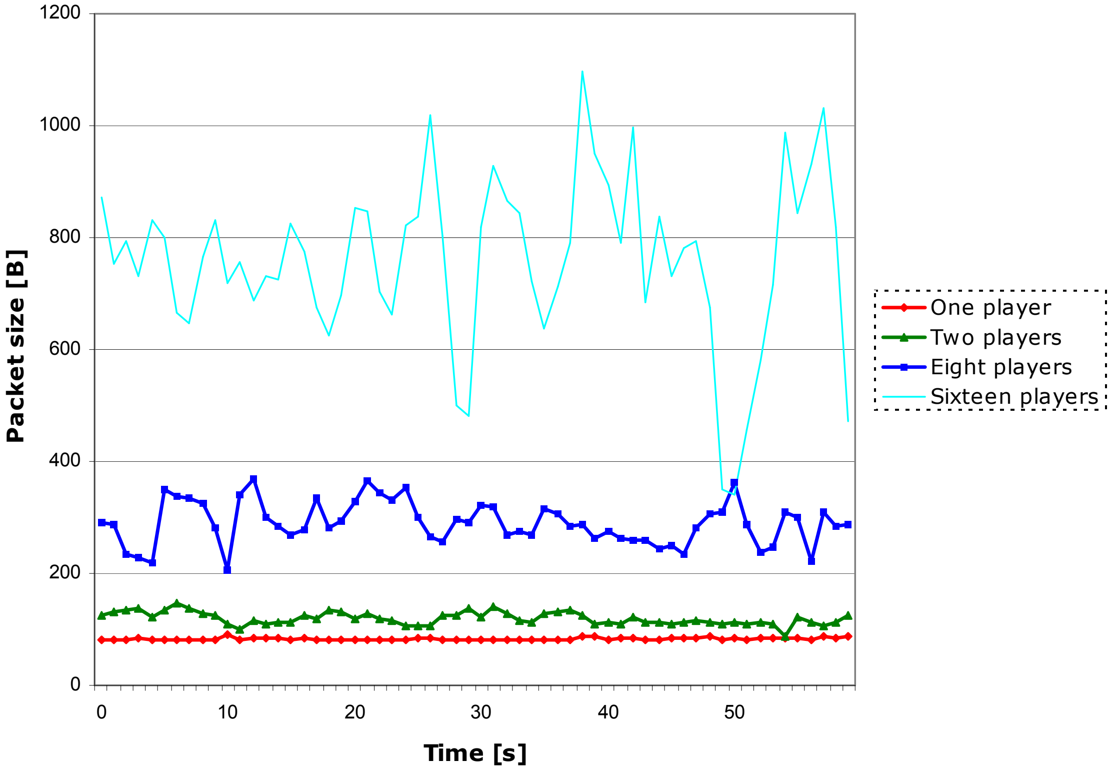

Figure 23. Quake 4 downlink packet size with varying number of players

The test from which both Figure 23 and Figure 24 were taken from was too small to be played by more than 8 players. The result is that the size of each game state update increased faster than if the game world was bigger. The spikes presented in Figure 23 could be explained by the short and intensely violent struggles that were fought at short intervals. Since the game world is increasingly smaller per user, the probability of interaction/conflict between users increase – which is readily apparent in Figure 23.

The maximum number of players allowed in a game session is often limited by the game but is often configurable from the server. Other limiting factors could be the map being optimized for a maximum or fixed number of players, the server's available bandwidth, server performance, or client hardware performance.

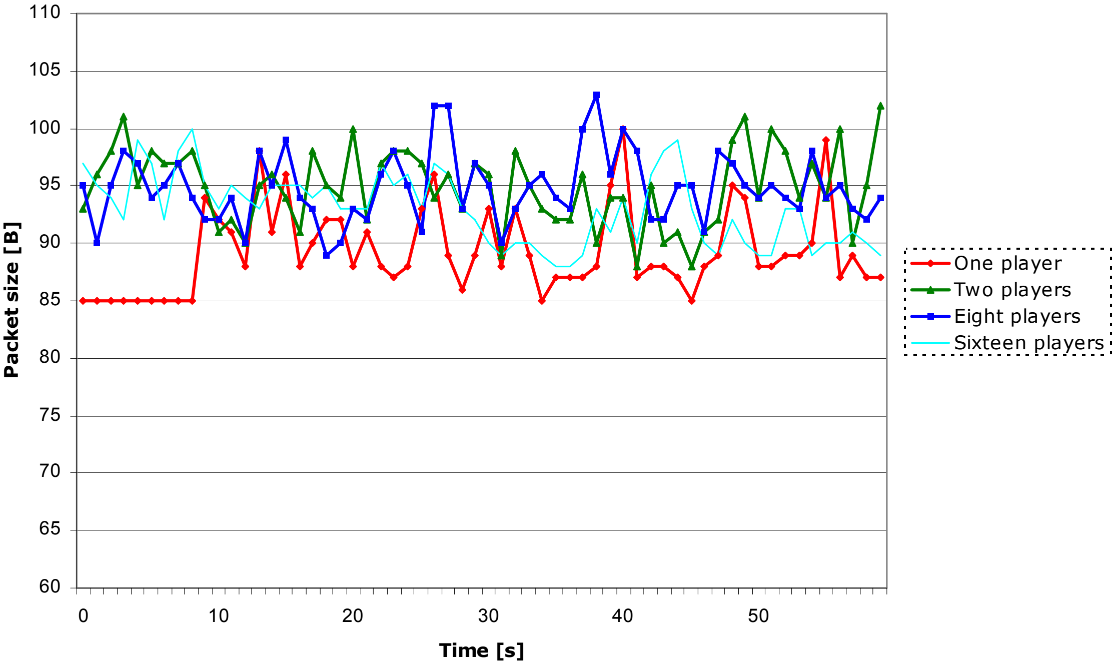

Figure 24. Quake 4 uplink packet size with varying number of players

5.7.2 Size of the virtual world

The size of the game server's representation of the world depends on the number of entities that exists in the game world. A bigger game world (e.g. map) will in most cases contain more game entities than a smaller map and will thereby place greater requirements on the servers' performance. On the contrary, a large map with N players may produce less interaction between the players, than the same number, N, players on a small map since their proximity will increase the number of interactions between the players (as noted in section 5.7.1).

5.8 Network performance

Together with client terminal performance network performance is the factor that has the highest impact on the end user's perceived online game quality. From a gaming perspective latency and packet loss is the worst villain when it comes to game quality impairments. Failing in support of the network throughput requirements usually results in failure of setting up an online game session, but once met not influencing the game to any appreciable extent. Packet loss, latency, jitter and throughput is described in the following sections.

5.8.1 Packet loss

As earlier described most network games uses some implementation of delta compression. As a consequence it is important that every game state packet arrives since it is not possible to unambiguously generate a new game state without the previous packets.

The main goal was to see if the introduction of packet loss and delay would separately or jointly have an affect on the traffic characteristics. One hypothesis was that the application, when it detected packet loss, would increases its sending rate to compensate for the reduction of received packets (i.e. sending duplicated packets). The hypothesis was found to be incorrect; no such behavior was seen with any of the tested games.

Renumbered packets here referred to packets that have a higher sequence number than expected. This is the results of one or more packets being delayed on their way to the destination. Depending on the implementation the packet will be queued to wait for earlier packets to arrive. Acknowledgements are continuously sent as piggy backed messages with the client's own update packets. Consequently the server will not retransmit packets that it got acknowledgements for.

5.8.2 Latency and jitter

Latency's main impact is not on the traffic; but on the game quality. The main problem latency causes is delayed responses and action that could easily turn into a source of irritation and frustration for the gamer.

5.8.3 Throughput

Network throughput is a measurement of how fast bits can be propagated through a network. A distinction is often made between sending (uplink) and receiving (downlink) throughput capacity. As the average user receives more information than he sends, there exist a variety of broadband connections that delivers greater downlink throughput than uplink throughput. From a network gaming perspective, this is not optimal since it is common that games sends more information than they receives (see section 5.5). This asymmetric distribution of available capacity in fixed networks is not a serious problem because of the comparatively low throughput requirements, but this is not the case with wide area cellular networks. A network game usually has a fixed basic throughput requirement enabling online gaming with a limited number of players on the server. The requirements on the downlink channel usually grow with the number of players present on the server, while the requirements on the uplink channel are kept fairly stable throughout the game session.

5.9 Measuring game quality

In comparison with determining the quality of other interactive multimedia services, measuring gaming quality is difficult. Currently there have not been any deeper studies made on determining gaming quality and the closest recommendation is ITU's P.911: "Subjective audiovisual quality assessment methods for multimedia applications" [45] . The main problem and difference with gaming is the lack of the direct interactive aspect introduced by multiplayer gaming.

To prepare for the discussion in chapter 5 I ask several people complete a game quality survey. I attempted to use test subjects that are frequent gamers and that use online gaming services frequently. The survey was divided into two parts; in the first part each person was asked to answer some basic questions about their gaming habits. The second part was of a more practical nature. Each person played the game "Quake 4" against at least one other test subject of the survey in a controlled network environment. The subject of survey was presented with 20 different network conditions. The order of the presented conditions was randomized so that no pattern could be distinguished (neither ascending nor descending). Between each network condition each person was asked to two main questions. The first question was how the person experienced the quality of the just played session and the answer was to be marked with a X on a ten centimeter wide line where a cross close to the left edge represented the worst quality and a cross close to the right edge represented the best quality. Second question was if the test subject would accept this quality for online play. The key idea of these surveys was to gain an understanding of the acceptance for network impairments and **not** to extract specific requirements based on statistically significant data. The results of these surveys form the basis of the results presented in Table 7. The survey form used to collect test subjects answers can be found in appendix A.

5.10 Network performance requirements for a satisfactory end user game quality

Network parameters such as packet loss, delay, jitter, and bit errors can cause reduced game quality for the end user. Small values of packet loss or delay will probably pass the user unnoticed, but when the impairments growing larger, the gaming quality can quickly be degraded to "unplayable".

From a network perspective it is of interest to guarantee acceptable end user game quality. Using subjective judgment a list of network requirements was produced that, if fulfilled, should guarantee a satisfactory end user game quality in the games we have tested. Given the limited statistical foundation for these requirements are not well founded rather they should be considered as a *suggestion* of what requirements real time network PC games may place on the network.

Table 6. Network parameter requirements for satisfactory gaming quality

Parameter:	Requirement:
Maximum packet loss frequency (one way):	4 %
Maximum delay (one way):	30 ms

In a network environment that meets the requirements in Table *6*, the network will not cause any perceivable online gaming impairment for the user of the tested games. The requirement presented above was not met by all online gaming connections on the Internet and still there are a great number of online gamers. A possible explanation is that gamers are tolerant to some game quality impairments. Different gamers have different requirements on their perceived game quality. Generally one could say that the more advance a gamer becomes, the pickier he or she will be about the perceived quality. Obviously gamers accept impairments to a certain degree, such that gamers perceive the game quality to be "good enough". In Figure 25 network latency tolerance is illustrated for three of the games tested. The lower boundary tells that at this latency the first signs of degraded game quality were detected. The upper boundary indicates that latency above this level generated a game quality such that it was perceived as no longer playable. Networks should strive to have a latency below the level which causes noticeable impairments, but at the same time it is unnecessary to optimize for extremely low latencies since it will not improve the perceived game quality.

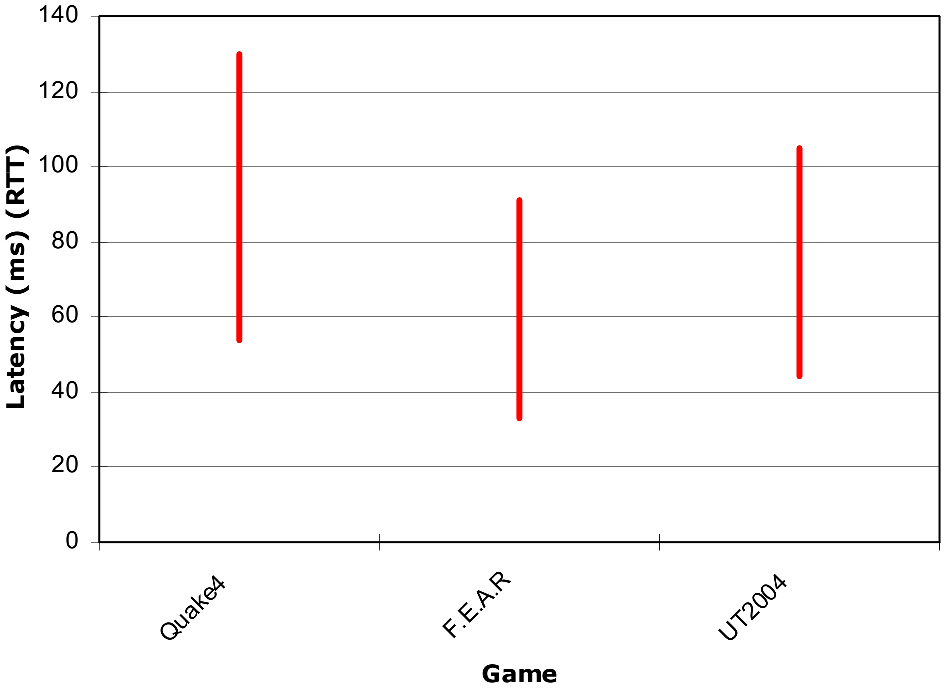

Figure 25. Per game tolerance to latency

When it comes to packet loss the tolerance was not easy to estimate. The general result is that a couple of percent is tolerable, but if critical packets (packets sent or received in highly interactive situations, i.e. when meeting an opponent face to face in a small room) are dropped, the consequence may still be considerable. It is hard to derive any precise conclusion out of the testing with packet loss when the losses in one session could lead to several packets being lost in critical situations while in the next session no packets were lost during the critical situation. A critical situation can be when two people see each other and begin to shoot at each other. To ensure fairness is such a situation it is important for both players to have a properly updated game world.

To get a better understanding of the requirements placed by hardcore gamers, some hardcore gamers were invited to participate in a game quality survey (see section 5.9 and the form in appendix A). During these questionnaires the game Quake 4 was used. The requirements presented in Table 7 is based on the answers collected during the survey and will, under the conditions present at the time for the survey, deliver a game quality that was perceived as acceptable by all participants.

Table 7. Game quality survey result: Performance requirements for Quake 4

Parameter:	Requirement:
Maximum packet loss frequency (one way):	8 %
Maximum delay (one way):	30 ms

A trend is that newer games have a tendency to tolerate more latency and packet loss, which is a result of more optimized network code. When the older game Quake 2 was compared with Quake 4 the difference was obvious. Already when 1% of the packets were dropped the consequences in Quake 2 was clearly noticeable, while this level of packet loss left Quake 4 unaffected.

5.11 VoIP services requirements compared to real-time gaming

Both the traffic characteristics and network requirements of a VoIP session vary because of differences between VoIP implementations because the different voice CODECs work in different ways. A comparison between VoIP and gaming services are interesting in many aspects. Voice conversion is generally viewed as the service that has the highest requirements on traditional mobile and the public switched telephone network (PSTN). Is it possible that real-time multiplayer gaming might place higher requirements on the network than voice services? If so, will a satisfactory game quality be achieved with a conversation QoS level or is there a need for an even more network demanding category?

There has earlier been a tremendous amount of study of voice traffic and the latency's impact on the perceived voice quality, e.g. in ITU-T. According to [56] 150 ms one way (approximately 300 ms RTT) delay will be sufficient to guarantee that most people will have a satisfactory highly interactive voice conversation in a echo-free environment. If we compare this number to the latency tolerance suggested earlier in this report (see Figure 25 and Table 7) 300 ms is far above the level where gamers found the game quality to suffer too much and therefore begin looking for alternative servers to play on. This statement is based on only a limited number of real-time multiplayer enabled game titles I examined, and is not necessarily valid for multiplayer gaming in general, but it should give a hint of the requirements for real-time multiplayer games compared to VoIP.

Chapter 6
Game traffic in cellular networks

With the increased capacities in cellular networks more and more services are both possible and (commercially) available. With the upcoming improvements of the 3G network and the new 4G networks, the available capacity is approaching today's wired broadband access networks capacity. Unfortunately the nature of cellular network is more complex than traditional wired networks. From a gaming point of view, even if the throughput is sufficient, latency and latency instability could preclude satisfactory multiplayer game quality. This chapter will look closer at gaming in wireless networks, examine what is possible today, and discuss what could be done to increase end user game quality. All traffic mentioned in this chapter has been either been generated and captured by me, or only been captured by me.

6.1 Cellular specific complications

The two main performance issues in mobile networks are latency and bandwidth. In wired packet switched networks packet loss is an issue with online gaming, but in the radio access network (RAN) packet losses are transformed into delay due to usage of link layer retransmission algorithms. Of course if the loss occurs before or after the RAN, the packet will just be lost. In general latency in today's mobile networks can be up to several seconds while the latency in wired networks is of the order of milliseconds. The available capacity per user is usually significantly lower in mobile networks than in fixed networks. Even if newer mobile network deliver higher peak data rates and lower latency, the nature of wireless connection are likely to be more unstable than that of wired networks. When you begin to move between cells handover/handoff delays are introduced which should be considered when designing real-time multiplayer game services. Some of these impairments could be the result of signal strength degradation or of increased number of users in the cell or of increased demands by one or more other users.

As a result of these less stable characteristics of a wireless network more work needs be done to be able to deliver reliable services compared with today's Internet connections. One important service that requires a reliable connection is voice traffic. Today more and more voice traffic is transported over IP-networks which increases the expectations from the end user of the network. It is easy to make the assumption that it will be easier to guarantee a specific level of service quality in a big wired network like Internet than in a cellular network due to the huge difference in available capacity, but this is not necessary a correct assumption. Most cellular networks are owned by a single operator or by a few operators. Mobile operators have in general good control of his network and the configuration of the equipment which increases the possibility of guaranteeing a specific level of performance from one end node to another. In the case of Internet the traffic often has to pass through equipment owned by company other than the access network, i.e. the destination node is not in the same ISP's network as the source.

6.2 Testing under optimal vs. realistic conditions

It is important to realize that the current network condition plays an important role for a cellular link's performance. A crowded network generally performs worse than an network where there are only a few users. Of course, this depends on the users' activity and the network equipment, but is generally true. During the tests in this thesis either (a) a network where I was, if not alone, it was assumed that I was likely to be the only user in the cell or (b) a public commercially available network where the conditions are more realistic but less controllable was used. For each test, I will try to clarify about what environment the tests were carried out in and distinguish whether or not the conditions could be viewed as realistic. Optimal conditions are interesting from several views. One is that it is possible to check whether a specific application or service is even possible with a specific technology. In my case, a question was if online gaming with acceptable game quality was achievable with the tested technology. On the other hand such test do not show whether the service/application can be introduced into a production/commercial network since the available capacity per user is not the same.

The realistic network in my tests was provided by a commercial Swedish operator covering Stockholm, Sweden. The tests were performed when the chances of simultaneous active users and traffic was high, e.g. mainly during working hours. The operator of this network will from now on be referred by the name "ABC".

6.3 Experiments with gaming in cellular networks

6.3.1 Network setup

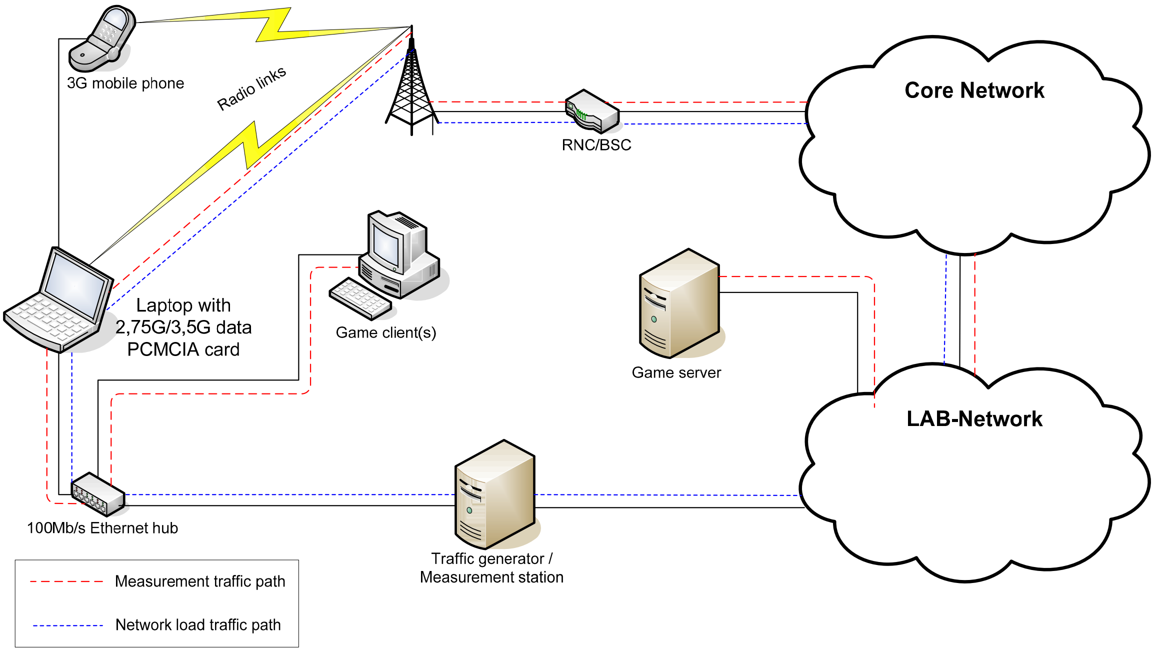

Figure 26. Cellular network test environment

Figure 26 is a simplistic picture of the network setup for the experiments in the controlled environment. To overcome the clock synchronization problem of measuring one way statistics the same computer ("Traffic generator / Measurement station" in Figure 26) was used both to generate and send network traffic through network interface connected to the LAB-network and to collect the traffic arriving on a second interface after it has propagated through the wireless link being tested. Due to having only a limited number of game clients, only three game clients where used at a time. Since every client was alone in their access network this limitation did not have any effect on the outcome of the tests, but may lead to a bias suggesting that there are no inter-client traffic effects.

6.3.1.1 GPRS/EDGE/HSDPA test setup

Due to some technical problems with the device drivers locking the mobile data PCMCIA card to regular (R99) 3G network, an alternative solution was used. This alternative solution consisted of a 3G-enabled mobile phone (a Sony Ericsson M600i) that was connected to the laptop through a USB 2.0 connection. Communication between the mobile phone and the computer were made in 12 Mbit/s. The mobile phone's built-in modem was used when establishing the data connection.

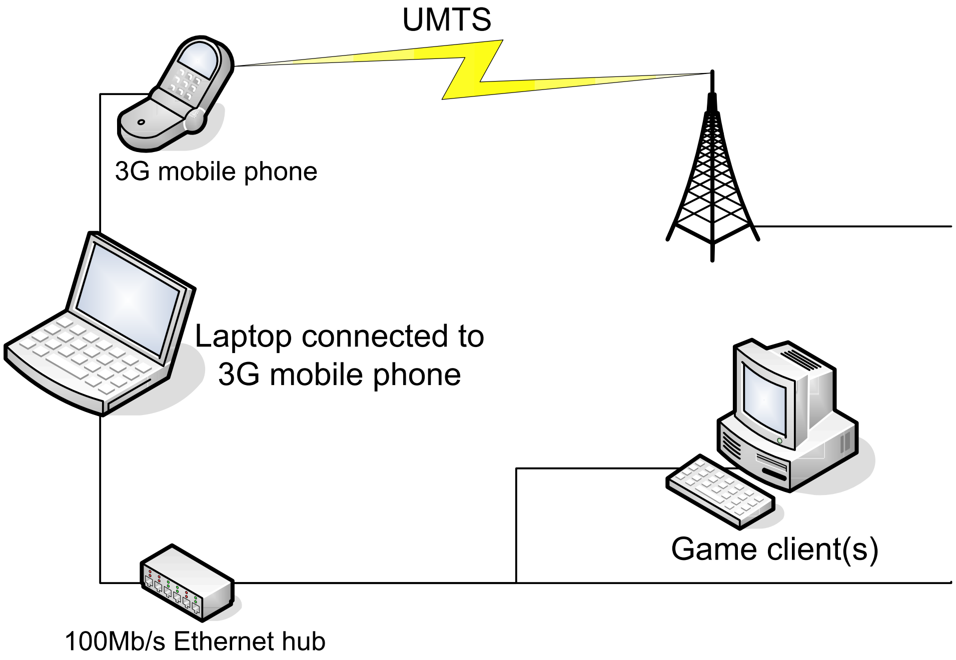

Figure 27. UMTS specific test setup

During the rest of the tests, a PCMCIA 3G-data modem was used to establish the data connection (both when connecting to operator ABC and when in the controlled cellular environment).

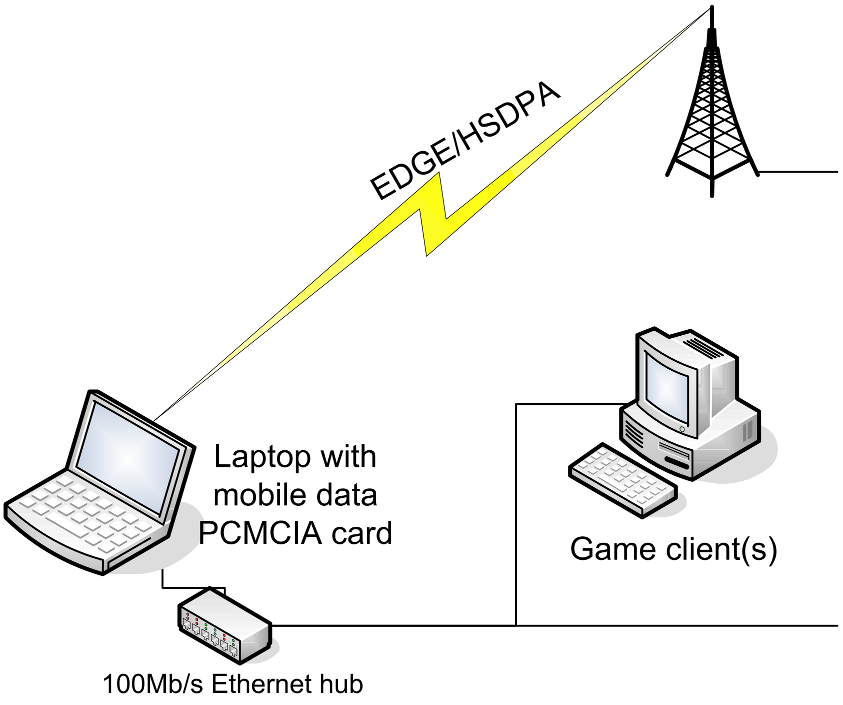

Figure 28. EDGE/HSDPA specific test setup

To understand the actual performance of each access technology some initial performance measurements were made. The first measurements used TPTest [46]. The results were then verified with file transfers between the game server and the game client. Subsequently both the round trip time and whether or not it was possible to play the game Quake 4 was tested. The results of the performance measurements in the controlled environment are presented in Table 8.

Table 8. Measured performance in the laboratory environment

Measured connection performance	**GPRS**	**EDGE**	**UMTS**	**HSDPA**
TCP downstream	42,47 kbit/s	211,78 kbit/s	353,31 kbit/s	1,44 Mbit/s
TCP upstream	22,79 kbit/s	99,55 kbit/s	57,27 kbit/s	355,15 kbit/s
UDP downstream	44,87 kbit/s	221,29 kbit/s	378,75 kbit/s	1,55 Mbit/s
UDP upstream	23,15 kbit/s	116,75 kbit/s	62,53 kbit/s	381,75 kbit/s
Approx. best case latency (RTT) (measured between MS and first node outside the core network)	200 ms	140 ms	103 ms	70 ms
Quake 4 multiplayer session possible? / With acceptable quality?	No / -	Yes/No	Yes/No	Yes / Yes

In the tests where the operator ABC was used the performance measurements are shown in Table 9. Due to firewall restrictions, an IP-tunnel (with the virtual tunnel software “VTun” [57] was used) was setup between the laptop in Figure 27 to a node within the LAB-network.

Table 9 measured performance in the ABC's network

Measured connection performance	EDGE	UMTS
TCP downstream	209,12 kbit/s	364,06 kbit/s
TCP upstream	81,10 kbit/s	57,73 kbit/s
UDP downstream	230,83 kbit/s	381,35 kbit/s
UDP upstream	110,65 kbit/s	63,94 kbit/s
Approx. Latency (RTT) (measured between MS and first node outside the core network)	240 ms	250 ms
Quake 4 multiplayer session possible? / With acceptable quality?	No/-	Yes/No

6.3.2 Network load's impact on latency

One important factor when playing games over any type of network is the latency between the client and the server. In wired packet switched networks the network load has less influence on the latency in a local network than it has in a wireless network. In general, the utilization of a wireless link plays a big role on the delay over this link. How the performance of the wireless link is affected by load differs between different technologies.

It is reasonable to assume that different types of load influence the network performance in different ways. With gaming in focus, the two most important network parameters for maintaining acceptable multiplayer gaming quality are latency and bandwidth. To investigate how the network's load influences gaming quality several tests with different kinds of load were performed.

In the first test the difference in performance between loading a network with small and large packets were studied. In the second test, a more realistic traffic load was used based on known traffic patterns. The third test examined how the game quality is affected by multiple simultaneous gaming sessions. To generate traffic originating from multiple game servers, a traffic simulator was used with parameters derived from the game traffic characteristics of the game Quake 4 collected during the experimenting with games in wired networks. The reason why Quake 4's traffic was chosen was because it representative of modern real-time multiplayer game within the genre of interest, its throughput requirements are similar to the other studied games, and this game sends packets with small sizes at a high rate similar to the other tested games.

The equipment setup illustrated in Figure 26 was used. The performance parameters for each network load configuration were: packet loss ratio (PLR), network latency, and inter arrival/departure times.

Table 10. Approximate throughput requirements for a successful Quake 4 multiplayer session

Approximate average throughput requirements for a successful Quake 4 multiplayer session	
Uplink:	50 kb/s (peak 56 kb/s)
Downlink:	21 kb/s (peak 48 kb/s)

6.3.2.1 GPRS

According to the results of these performance tests, the GPRS uplink performance is not close to the level Quake 4 requires for problem free multiplayer sessions. As shown in Table 10, Quake 4 will require a bandwidth of approximately 50 kb/s uplink and 20 kb/s downlink while GPRS; delivered around 23 kb/s uplink and 45 kb/s downlink. Therefore GPRS did not have sufficient performance to meet the game's basic performance requirement, hence not additional studies of this network were made.

6.3.2.2 EGPRS/EDGE

As can be seen in Table 8, EDGE offered a throughput performance similar to the UMTS mode of the 3G network. In my configuration, EDGE offered even higher uplink bandwidth than the 3G R99 setup did. During the tests with the laboratory cellular network, shorter periods of time with performance degradations could be observed. In Figure 29 and Figure 30 these degradations are showed as spike(s). The reason for the high peak values of the packet loss spikes is that during the time the network performance was low, almost no packets reached the destination and because the measure period was limited to two minutes per load setting the resulting packet loss period had a larger impact on the ratio.

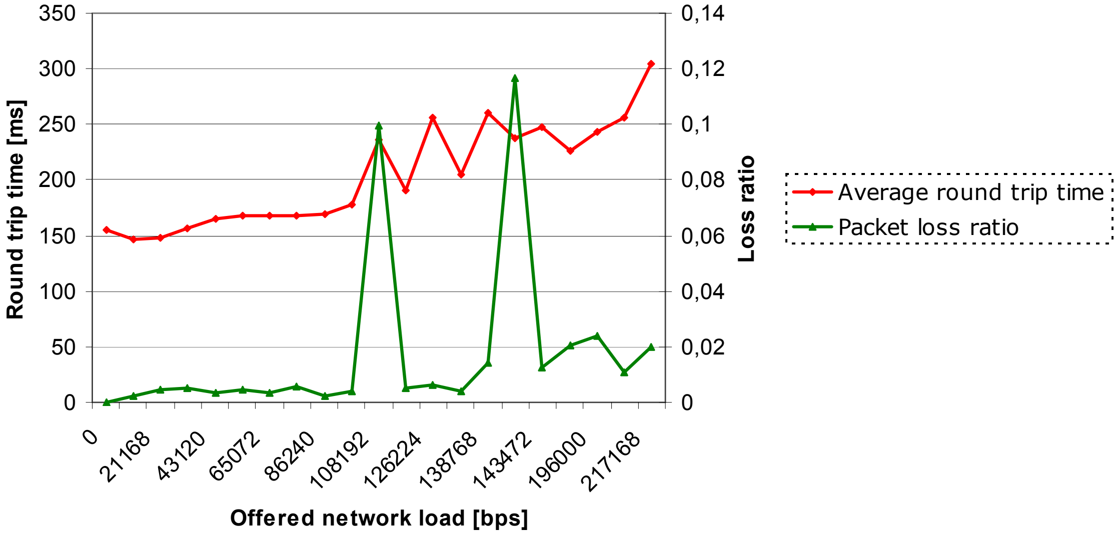

Figure 29. Small packet traffic's influence in an EDGE network

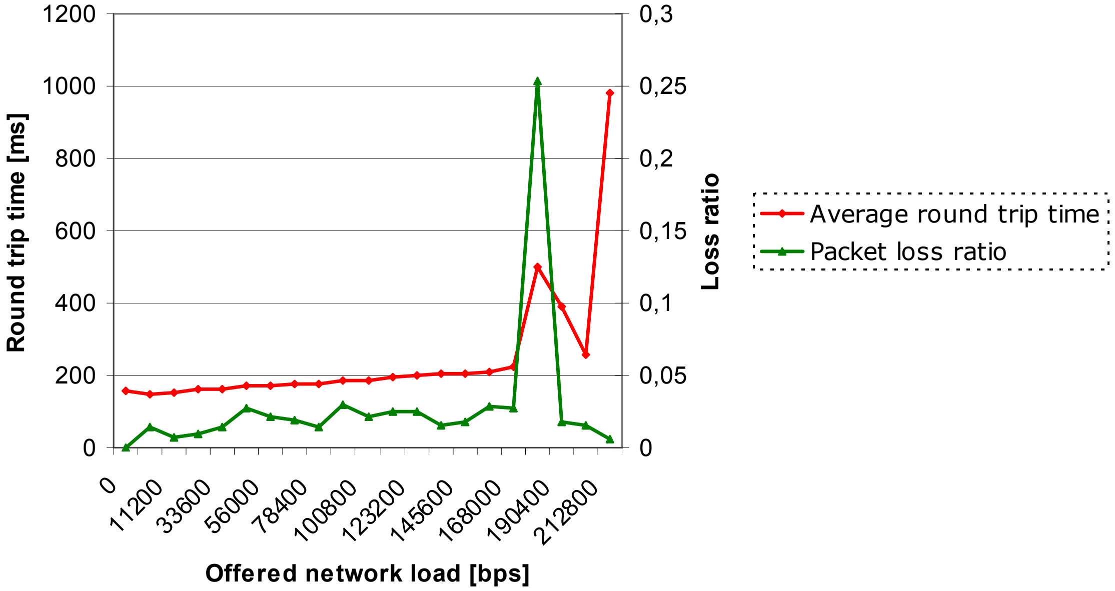

Figure 30. Large packet traffic's influence on in an EDGE network

6.3.2.3 UMTS

The practical maximum capacity of the UMTS link was up to 384 kb/s both up- and downlink. Since it is takes a lot of resources to dedicate 384 kb/s channels to users, each cell is usualy configured to hand out only a limited number of 384 kb/s channels and then hand out channels with lower bitrates. Additionally the uplink is usually capped to a value lower than 384 kb/s. During the experiments in the laboratory environment performance of close to 384 kb/s downlink and 64 kb/s uplink were measured.

The same behavior with small periods with no activity also appeared in the UMTS network as earlier in the EDGE network. In Figure 31 and Figure 32 show how the average round trip time is influenced by the offered network load. During these experiments the setup illustrated in Figure 27 was used. In connection with the experiments a Quake 4 game session was successfully established with the server. Unfortunately the game session suffered from serious game quality impairments, such as highly delayed responses and jerky motions.

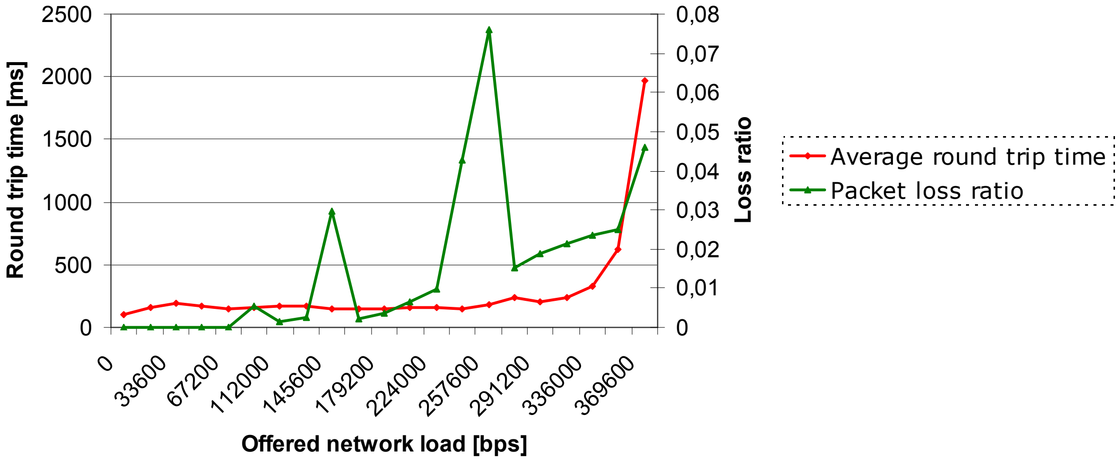

Figure 31. Large packet traffic's influence in an UMTS network

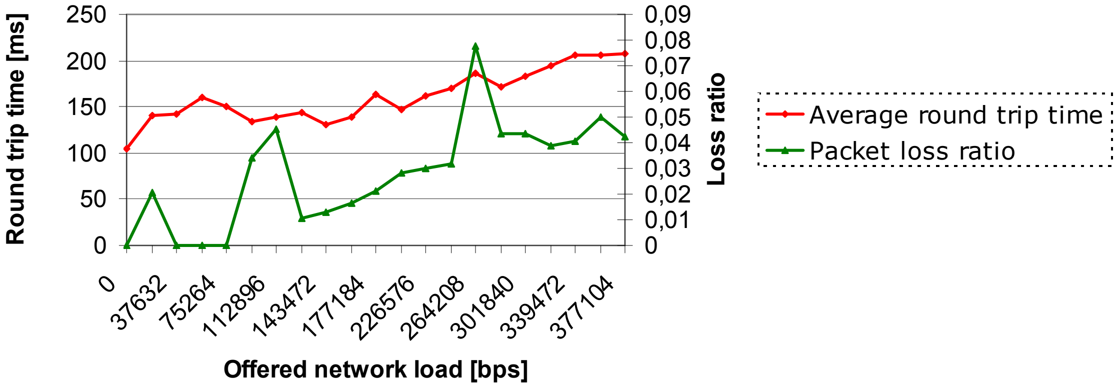

Figure 32. Small packet traffic's influence in an UMTS network

6.3.2.4 HSDPA

HSDPA is a big step forward for gaming over cellular networks. This is most obvious when looking at the round trip times, but is also noticeable when it comes to capacity. In Figure 33 the latency variation is shown for up to nine simultaneous game sessions over a single HSDPA connection (one player per game session). The number of simultaneous game sessions in my test was limited to eight sessions simply because the uplink bandwidth limit was reached, but it is interesting to notice that the round trip times were rather stable until the uplink capacity was reached. A scenario with multiple simultaneous game sessions may not be of interest for a mobile handset, but none the less interesting when it comes to mobile broadband since one HSDPA connection could then be used by several end users simultaneously.

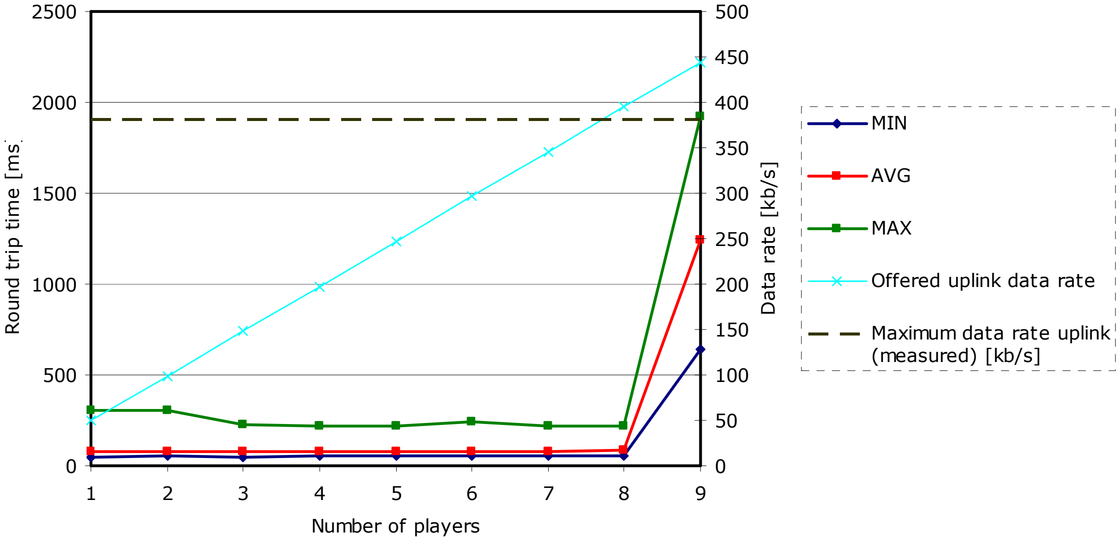

Figure 33. Multiple gamers influence on the latency over a single HSDPA connection

As mentioned earlier multiplayer game quality depends not only on latency and available throughput, but also on low packet loss ratio and steady inter-arrival times. In Figure 34 a subjective determination of the game quality was made for one of the clients in conjunction with the results presented above. If the game quality scale in Figure 34 is compared to that of Table 2, a game quality level of 4-5 on the game quality scale corresponds roughly to good quality, level 2-3 to acceptable, and level 0-1 as unacceptable.

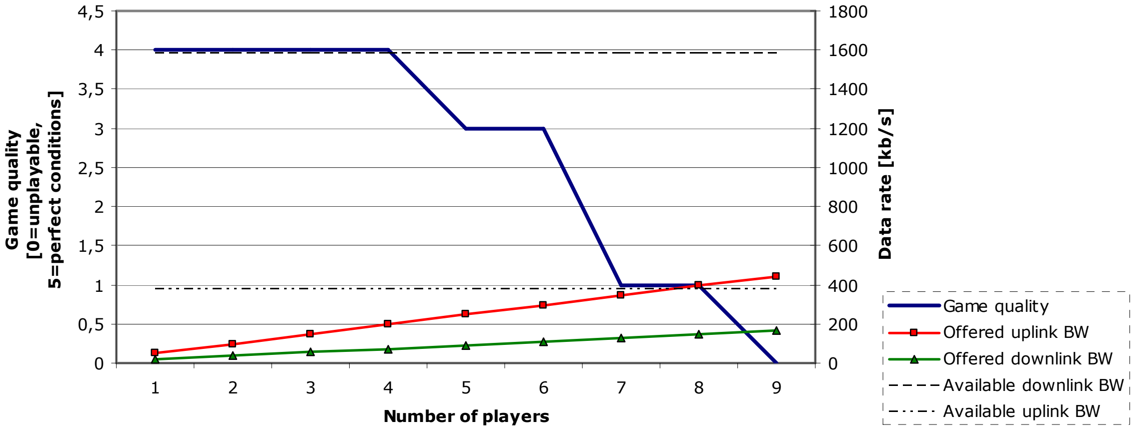

Figure 34. Multiple gamers influence on the game quality over a single HSDPA connection

If we assume that a game quality level of three (where zero representing a unplayable quality and five perfect quality) to be a level where most players still accept the quality, a single HSDPA connection could carry about six simultaneous game sessions.

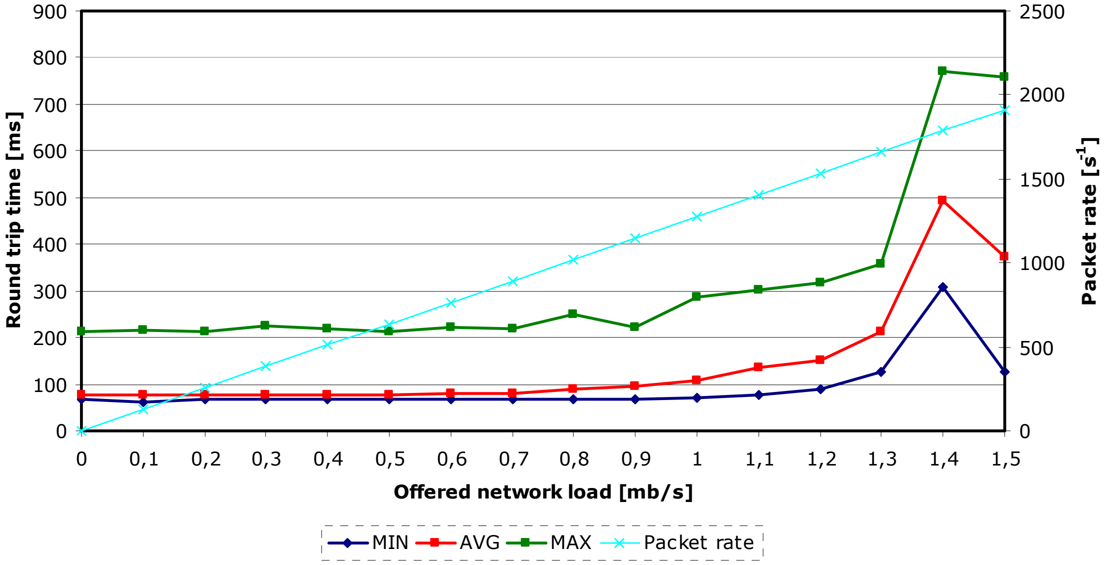

Figure 35. Small packet traffic's influence in an HSDPA network

HSDPA also shows a more steady behavior, from a latency point of view, compared with the other technologies studied and did not show periods of inactivity, unlike that experienced with both UMTS and EDGE (Figure 35 and Figure 36).

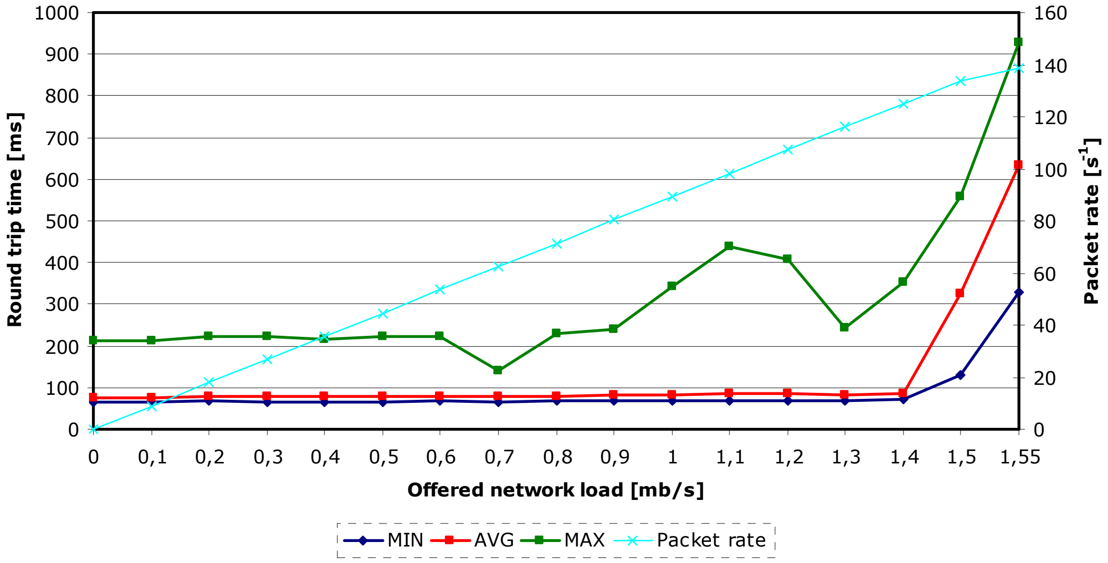

Figure 36. Large packet traffic's influence on latency over a single HSDPA connection

6.4 Gaming in a WLAN environment

With the popularity of using WLAN at home and the increased number of hotspots available to the public, online gaming through WLAN connectivity has become more and more popular. Today there exist two major portable terminals that provide multiplayer gaming over WLAN; the *Sony Playstation Portable* [19] and its competitor *Nintendo DS Lite* [20]. From a real-time gaming perspective this is interesting since their services are working today and are perhaps the most obvious challenge to portable wide area cellular terminals and mobile gaming.

Another fact that makes gaming over WLAN interesting is that several PCMCIA cards designed to enable laptop users to connect to GPRS/UMTS services also provide WLAN connection possibilities. Additionally, most laptop and PDA manufacturers build these interfaces into their products. Even if the games played on these terminals may not be create high real-time requirements, they are moving network infrastructure vendors and network operator towards a more game aware future.

One main problem with wireless systems in general and WLAN in particular is that their performance degrades with the number of users and the load on the base station. Such performance degradation affects throughput, latency, and packet loss. Since low latency is vital for online gaming, the load on a wireless network could be crucial for the gaming quality and experience.

Two experiments were performed regarding gaming quality in WLAN environments. Both tests examined how network load, consisting of either large or small packets, on an IEEE 802.11b access point influenced the gaming quality with regards to latency. This will indicate how stable the latency in a hot-spot will be when loaded with either large packets symbolizing file downloads or small packets which could be originate from either game clients or from VoIP clients. The results will give an idea of the relationship between the load on an access point and the latencies perceived by the wireless nodes. Comparing with earlier results regarding game quality requirements gives a rough estimate of how the game quality would have been influenced and perceived.

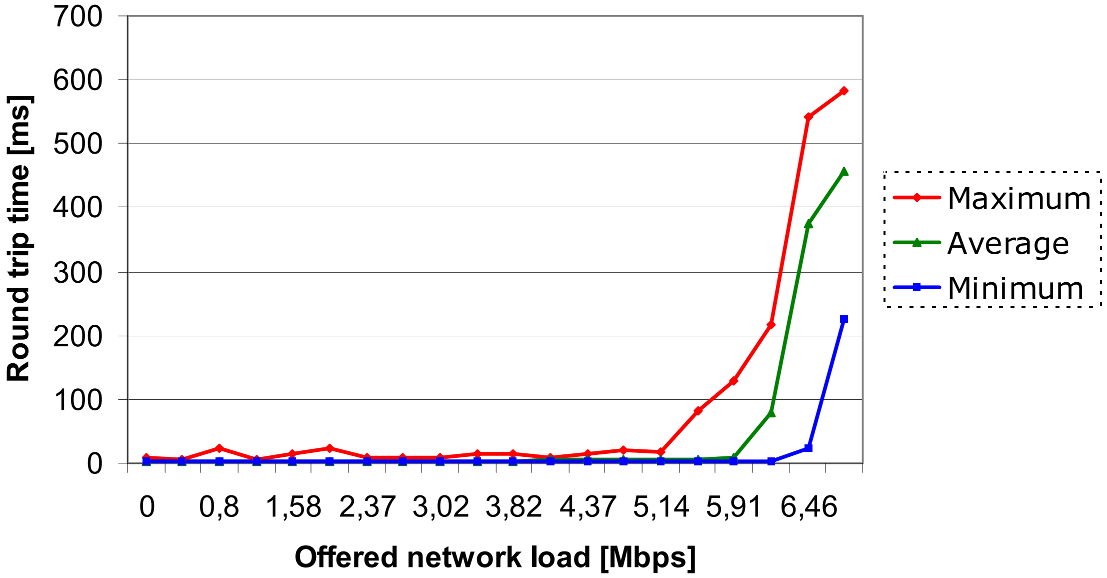

Figure 37. Large packet traffic's influence on an IEEE 802.11b access point

As indicated in Figure 37, the increasing load with large, 1400B packets, results in a stable RTT until the offered network load is close to the access point's maximum throughput. If only the latency is taken into account, most real-time games will be playable with acceptable game quality even with fairly high load on the access point.

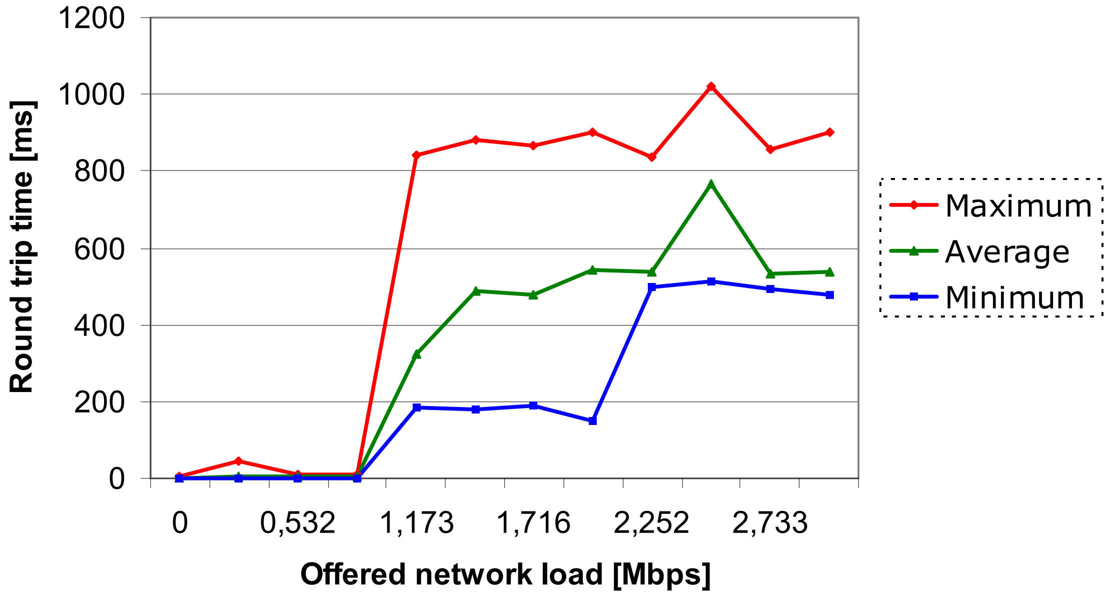

Figure 38. Small packet traffic's influence on an IEEE 802.11b access point

Figure 38 shows the results of the same experimental setup with the difference that the network load consisted of small packets (which could represent a large number of online gaming sessions or VoIP calls). A drastic difference is seen compared to Figure 37. The latencies experienced by the nodes connected to the access point raises above 200 ms even before the offered network load has reached 1 Mbit/s.

The main underlying cause to the difference in measured RTT between Figure 37 and Figure 38 is increased queuing delays caused by the limited maximum packet rate in the wireless network. According to [62] a simple approximation of the maximum packet rate could be done with equation (2), where R_{MAX} is the estimated maximum packet rate and T_i is the overall frame transmission time per host in a network with N active hosts.

$$R_{MAX} = \frac{1}{\sum_{i=1}^{N} T_i} \qquad (2)$$

The overall frame transmission time could be described as the sum of the time it takes to send the frame, the propagation time, the overhead, and the expected average resend time that is dependent on the proportion of experienced collisions. With a decreased packet size, the time it takes to send a frame will be decreased, the expected number of collisions will increase (assuming multiple active hosts) while the propagation time and the overhead (consisting of time added by the distributed inter frame space (DIFS), the short inter frame spaces (SIFS), and the acknowledgement) will be unchanged. The resulting maximum frame rate will increase when the sizes of the packets are decreased from 1400B to 98B, but the change in packet rate is not linear with the change in packet size.

6.5 Portable game terminals and multiplayer games

Among the current generation of portable game terminals that provide multiplayer possibilities, WLAN is the most popular and common wireless interface. This is due to several possible reasons. The first reason is that WLAN is today's most spread technology that enables two nodes to talk to each other with high band width (usually greater than 2Mbit/s of actual throughput). Additionally, infrastructure mode can be used to connect multiple devices to an access point. The biggest limitation of WLAN is that the maximum range between two nodes is 30-100 meters depending on the physical environment.

With the growing demands of online gaming services, a natural development is to extend game terminals with cellular data modules. The most obvious reason why this has not happen yet is the, in many aspects, poor performance of today's cellular networks and the fact that most operators charge for data per kilo byte. According to [52] this is changing and more operators are moving towards so called "flat-rate"—where the customer is charged a fixed rate monthly instead of per unit of data transferred (although, sometimes the operator may place a maximum limit on the data transferred each month).

A conceivable continuation, from a technical view, of the development of portable game terminals would be to utilize cellular networks. A next step from today's terminals would be the introduction of mobile network modules integrated into the next generation terminals to enable both Internet connectivity and online gaming services. Not only will this enable end users to play online almost wherever they are, there may be additional revenues to collect for both network operators and for network vendors in the form of demand of both gaming enabled equipment and premium services that give priority to game traffic. From an operator's point of view gaming services should be considered attractive since they will not generate significant traffic (see Table 12 regarding the traffic generated by games). The main requirement is bounded low end-to-end delay, which could be guaranteed by using QoS to give game traffic high priority. There are several possibilities for game services; either the operator could use prioritization of game traffic as a competitive advantage or offer game traffic priority as a premium service.

Current portable game terminals are, as mentioned, equipped with WLAN connectivity and according to Marc Claypool [43] the Sony PSP game traffic bit rate is around 300 Kbps or less with the games he tested. If we compare the results of his studies with the result of the PC game traffic measurements in chapter 5, it is clear that these portable game terminals generate a lot more traffic than PC games. During their tests, the game terminals were communicating with each other in a peer-to-peer fashion, unlike the other available mode, infrastructure mode, where the game terminals are communicating through WLAN-access points. The amount of traffic generated may be explained by the fact that WLAN has a rather high bandwidth within its short range, thus game developers have no need to optimize their communication protocol – as traffic throughput is not a problem. Of course the traffic a terminal generate could be based on the medium used as a carrier, but since today's high-end cellular network theoretically support the same bitrates used by the Sony PSP, it could be possible to have a satisfactory Sony PSP game session over a mobile data link today. Of course the work needed to modify a Sony PSP to natively support a 3G cellular connection is beyond the number of working hours allotted for this thesis. However an experiment was performed to determine if it is possible to have a successful Sony PSP multiplayer game session over a HSDPA connection. In addition should be mentioned that when the Sony PSP games are using infrastructure mode its traffic differs significantly from the traffic generated in the peer-to-peer mode (see Table 11). The main reason is that games using infrastructure mode generate smaller traffic volumes is because they are mainly designed for online game services and therefore optimized for broadband connections unlike earlier described peer-to-peer.

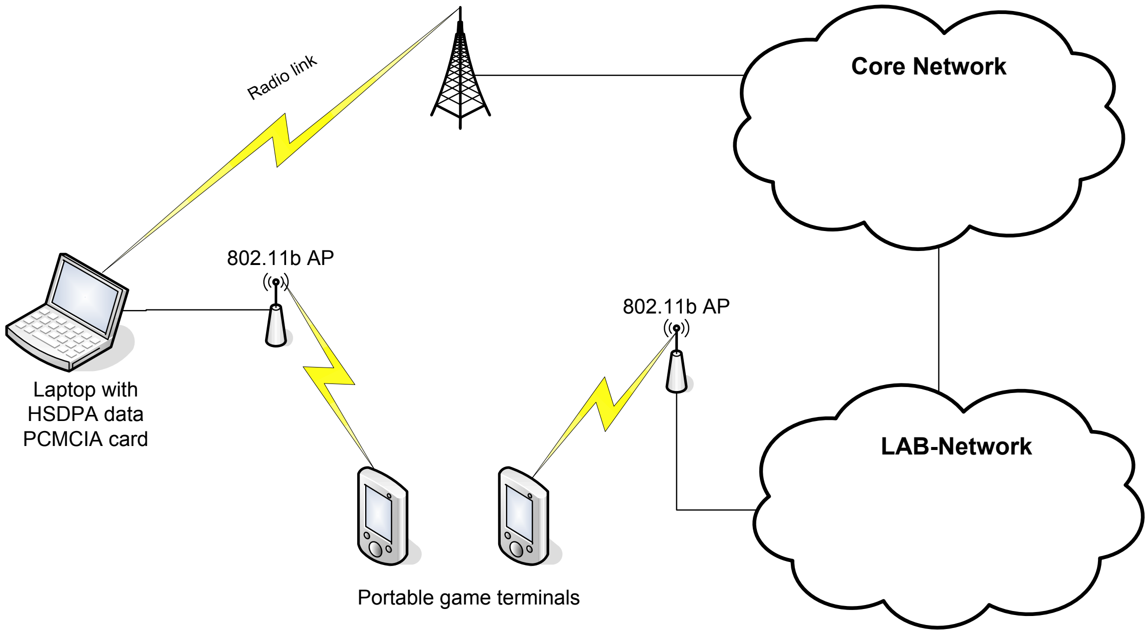

Figure 39. Laboratory setup for establishing a PSP multiplayer session over HSDPA

To enable the Sony PSP to play over HSDPA each terminal was connected to a local IEEE 802.11b access point. The first access point was connected to the LAB-network and the second access point was connected to a laptop which utilized a HSDPA connection to connect to the

LAB-network. So that the game terminals thought they were on the same local area network, an IP-tunnel was established between the laptop and a server within the LAB-network. The setup is illustrated in Figure 39.

Table 11. FIFA 2007 (PSP) Traffic characteristics

PSP FIFA 2007	Avg. Packets/s	Avg. Packet size	Avg. bit rate
Uplink	13,661 s^{-1}	114,750 B	13 kb/s
Downlink	13,613 s^{-1}	113,515 B	12 kb/s

The result of the test is that the Sony PSP game FIFA 2007 by all means is playable over a HSDPA link. Since FIFA 2007 was performing so well over the HSDPA link, an additional test was carried out to verify whether or not an EDGE connection would deliver sufficient performance. Even the EDGE connection proved to deliver acceptable game quality.

6.6 Methods to optimize games for cellular environments

To gain better gaming performance you could make changes to the network or the terminal or to both. A user can not make changes in the cellular network; the main possibility for users and game developers to boost online game quality is either to tune their multiplayer game configuration or to develop better performance multiplayer gaming protocols that are optimized for wide area wireless networks. A justified question is whether this kind of optimization is really needed since coming cellular networks offer better capacity and lower latencies. Today’s mobile networks are not used as a broadband extender and the game traffic generated from stationary game consoles in today’s mobile network are minimal. However, in coming years fixed point mobile connections will be used as an end user broadband service in areas not covered by the wired broadband services. One attractive Internet activity today is online gaming, and these kinds of services will most certainly also be desired by the users of mobile broadband.

6.6.1 Game configuration optimization

The majority of multiplayer games allow the user to configure a large number of parameters that affect the game and to some degree the multiplayer game traffic. To simplify the configuration process it is common that predefined game configuration profiles are delivered with the game, the user simply activate the game by choosing the relevant network access method (dial-up modem/xDSL/cable modem/etc.). For a more advanced user, there are many game parameters that could be tweaked. One example is Quake 4 which has more than 800 manipulateable variables [47] that in different ways influence the game. These variables could be altered either by changing in the game's configuration files or by enter the changes directly through a console during game play.

6.6.2 Game design optimization

Game design optimization could optimize game traffic for the network changing the way that the clients and server talk to each other. In the beginning of the online game era, the main Internet access type was through traditional modems with a maximum data rate of either 28,8 kbps, 33,6 kbps, or 56 kbps. Most games that were developed during this period of time had their multiplayer services optimized for such kind of connections. Nowadays there exist a greater variety of Internet access connection possibilities and the mainstream user is not longer using traditional modems to connect to the Internet. This has also been reflected by game developers' minimum requirements for online gaming being some kind of broadband connection.

Today's PC-games are neither intended nor optimized to be played over mobile links, but mobile packet switched network characteristics have many impairments common with those game developers today struggle to conquer. As proven in earlier experiments in this chapter techniques like dead reckoning (see section 4.6.1) and predictive contracts (see section 4.6.2) work well over mobile links. Of course such methods could be more effective if they were optimized for cellular networks, but these methods are already fulfilling their purpose, to hide latency and network impairments from the gamer. The mobile real-time gaming quality of today might not even be playable, without the technologies developed for the real-time PC online gaming. An open question is if it is possible to take the techniques developed for PC games and optimizes them for a mobile environment.

6.7 IMS

The IP Multimedia Subsystem (IMS) was originally an attempt by the cellular network industry to control and take advantage of new and upcoming IP based services. Since gaming traffic puts real-time requirements on the network, there is a interest from both end users, to get a service with acceptable quality, and from network operators, to have satisfied customers, to prioritize gaming traffic before other traffic. With regard to low latency, gaming traffic may place higher real-time requirements on the network than VoIP; thus gaming might require a QoS class different than currently designed for real-time multimedia services.

Chapter 7
Economic and business opportunities in an online network gaming market

Driven by the increased broadband penetration the online gaming market has grown tremendously over the last few years. The value of the global online gaming revenues are estimated to grow 40% from \$3.5 billion in 2005 to about \$5 billion in 2006 [26]; while the online gaming subscriptions within the Europe, the Middle East and Africa (EMEA) area are predicted to rise from \$6.0 million in 2006 to \$18.0 million by 2010 with an average monthly subscription costing around \$13 per subscriber [52]. This chapter gives a short overview of the current online gaming market.

7.1 Online gaming market overview

A clear tendency is that online gaming has moved from being simply a phenomenon within the PC game market to enter the console market, cell phone market, and the portable game terminal market. The online gaming market can be divided into several areas; console market, handheld game terminal market, PC game market, and cell phone market. Another clear tendency is that, despite the continuation of broadband growth, PC game popularity, which is still the dominant online gaming platform [52], is declining; it is expected that the console market during 2007 will be dominated by the introduction of Sony's and Nintendo's new game consoles [52]. Due to competition these consoles are trying to deliver the best online services possible; this will affect the market for hand held devices and cellular handsets with gaming possibilities – perhaps leading to a possible convergence between online services offered for the console market and for the handheld game terminal market.

With the creation of the online gaming market comes submarkets with services such as providing game content or online game communities. Another interesting submarket that online gaming has created is the use of dynamic advertising in online games. One company that has been active in this area is Massive Inc. Its video game advertising network allows advertisers to make their advertisements visible to players through online game services [53].

Within the online gaming market a variety of pricing and charging strategies has been seen. Most common is that you either pays for the game software and the online services come free of charge or the software comes for free and online services require a monthly or yearly subscription fee. In the later case, single player functionality is missing and the only way to utilize the game is through online services.

In the cellular telephone handset market the most popular games are those that come preloaded in the handset [52]. In recent years more and more phones are Internet ready and capable of downloading content, this opens up the market for selling games directly to the user of a cellular telephone handset, but the popularity of the preloaded games will presumable still be high until the carriers start offering subvention for handsets without preloaded games to force their users to actively download games. A risk that could hit online gaming on mobile phone handsets is that it will be negatively impacted by the attempts which carriers to have made to generate as high revenues as possible out of music and video services. For the operator gaming revenues could come from traffic generated, but they might also take advantage of mobile gaming portals and additional services around each portal. An advantage of gaming on mobile handsets is that the charging procedure is somewhat easier since small amounts could easily be added by the carrier to the subscriber's bill either as a one time fee or as a subscription fee.

7.2 Game distribution

The classic way to distribute computer game is on physical medium (i.e. on floppy disk, CD, or DVD) via specialist retail stores. This has for long been the most practical way to distribute games due to their usually relative large size and the low consumer bandwidth. Today as more and more end users possess high bandwidth connections, online game distribution has become an attractive alternative to traditional distribution. The main advantage for game developers of online game distribution is that they cut out the middleman, as game copies are sold directly to the end users instead via a publisher.

7.2.1 Digital game distribution

It is already common to distribute digital entertainment content via mobile networks. Distributing games via mobile phones could utilize the phone's built-in WAP browser and the payment could be made either by charging the user's cell phone subscription or by SMS fees (i.e. the user sends an SMS to a specific number and the reply is an activation code used to start the download of the desired content).

A successful digital content distributor is Valve's content delivery platform Steam [32, 33]. The main goals of their Steam platform is not only to deliver new games, but also to update games with new patches, enable communication between gamers, and help gamers find game servers. Through Steam users can purchase available games with credit cards. Game patches are, when possible, automatically propagated to the users in small chunks in the background so as to avoid disturbing the user's other activities. In situations where changes have an effect on both servers and clients (i.e. updates that force clients to play on servers with new server software) the patch must be installed before the player can utilize the server.

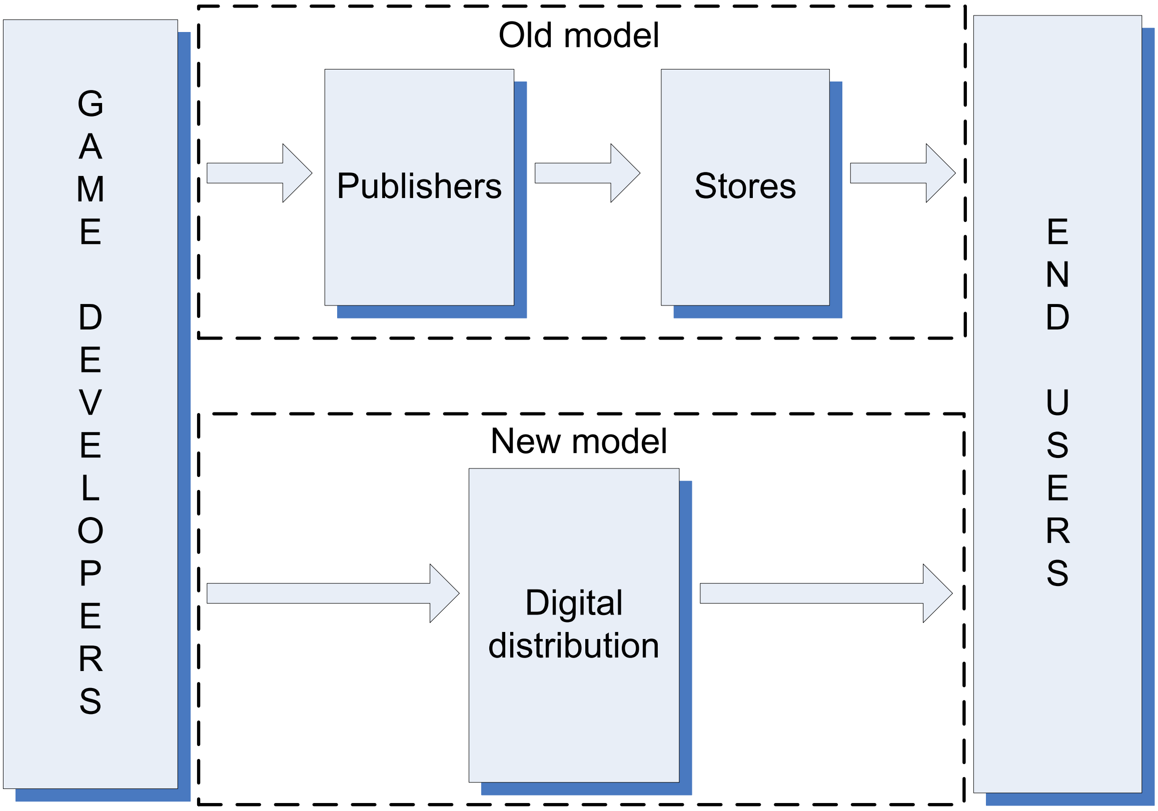

Figure 40. Alternative game distribution schemes

Chapter 8
Conclusions

What can be seen in the results of chapter 5 and 6 is that online gaming should be considered as a network application whose service quality is vulnerable to network impairments. Performance in mobile networks is in general more complex than performance in wired packet-switched networks in the sense that there are more causes for performance degradations in mobile networks than for wired packet switched networks.

An interesting development is the possible convergence of gaming services between different kinds of terminals. A conceivable outcome could be game services will be available to different game terminals on the same conditions, whether you are mobile or sitting in your living room with your console. To attract all the potential gamers to a service it is important to make it easy for the casual gamers **and** to support hardcore gamers (i.e. through support for their higher demands on lower latencies). Another interesting point was a comment during the game quality surveys: gamers that began their gaming career during the dial-up modem era probably have slightly lower requirements, since they already have experience of playing under "bad" conditions, than gamers that have only played over (home) broadband connections. If there is some truth in their statement, then the result may be increasing requirements since more and more people begin their gaming over broadband connections.

In this thesis games utilizing prediction models and games that do not have been studied. From a development point of view more and more games, on all terminals, will utilize their increased processing capacity to use prediction models, as we have seen the game quality improvement that this can bring. Network impairments are hard, if not impossible, to prevent. Network vendors should actively work to support real-time multimedia services and multimedia services, such as gaming. This means that they should actively work to improve the network impairment hiding algorithms in order to increase the tolerance to network impairments (such as jitter and packet loss).

8.1 Game traffic summary

Table 12. Summary of average game traffic characteristics per session

Game traffic summary	Packet rate [s^{-1}]		Packet size [B]	
Game title:	Uplink:	Downlink:	Uplink:	Downlink:
Battlefield 2	24	21	107	480
DOOM 3	61	14	104	144
F.E.A.R	26	19	84	73
Quake 2	66	21	84	174
Quake 4	63	15	98	169
UT2004	36	32	71	70
Average	**46**	**20**	**91**	**185**

The PC games studied generated traffic consisting of small packets, with sizes around 100B uplink and 200B downlink, at a steady rate of around 20 packets per second downlink and 50 packets per second uplink.

8.2 Mobile gaming forecast

It should first be mentioned that my approach is based on a technical point of view. The market is a lot more advanced than what I make pretence of with forces that counteract the technical development (e.g. companies that today make revenues through old-fashioned business models).

The evolution of handsets and cellular networks is important for both vendors and operators. It is of great importance to identify new opportunities to continue growing. As of today, available cell phone handsets do not deliver game quality that is good enough to attract hardcore gamers [22]. While casual gamers that to not make as high demands on their gaming experience as hardcore gamers (and therefore may be pleased with today's handsets) are not willing to put as much money in to their gaming as hardcore gamers do. An impediment to attracting casual gamers is that there already exist free or low cost handset games that meet the needs of these casual gamers. The big change will come when hardcore gamers find the wide area cellular handsets attractive enough for serious gaming. Operators and vendors demand to meet this coming need with both handsets and networks that can maintain as high an online gaming quality as possible. This will likely enable operators to generate revenues from guaranteeing a good gaming connectivity and from data access. From a network infrastructure vendor's point of view, gaming will create demand and therefore be a contributing factor that drives networks toward lower latencies and cell phone (and gaming terminal) vendors to develop more advanced handsets (which in turn enables more advanced services).

8.3 Wired network gaming forecast

In the coming years there will unlikely be any drastic network requirement changes caused by online gaming. The continuous development of hardware components will offer game resolutions corresponding to the HDTV standards enabling games to be experienced on HD displays. Higher resolution will enables a greater detail in the picture.

Terms and abbreviations

Term or abbreviation	Meaning
2G	2nd generation of mobile networks
3G	3rd generation of mobile networks
3GGP	3rd generation partnership project
CDMA	Code Division Multiple Access
DIFS	Distributed Inter Frame Space
EDGE	Enhanced Data rates for GSM Evolution
EGPRS	Enhanced GPRS
EMEA	Europe, the Middle East and Africa
FEC	Forward Error Correction
FPS	First Person Shooter
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communications
HD	High-Definition
HDTV	High-Definition Television
HSDPA	High Speed Downlink Packet Access
IRC	Internet Relay Chat
ITU	International Telecommunication Union
ITU-T	Telecommunication standardization sector of ITU
J2ME	Java 2 Platform, Micro Edition
MMORPG	Massive Multiplayer Online Role Playing Game
MTU	Maximum Transmission Unit
NTP	Network Time Protocol
PDU	Packet Data Unit
PSTN	Public Switched Telephone Network
QoS	Quality of service
RAN	Radio Access Network
RTS	Real Time Strategy

RTT	Round Trip Time (often measured in the order of ms)
RPG	Role Playing Game
SIFS	Short Inter Frame Space
SMS	Short Message Service
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunication System
WCDMA	Wide Code Division Multiple Access
WLAN	Wireless Local Area Network

References

[1] Roger Delano and Paul McFarlane. *Network Software Architectures for Real-Time Massively-Multiplayer Online Games*. McGill University, Montreal, Quebec, Canada, February 2005.

[2] Honghui Lu, Björn Knutsson, Margaret Delap, John Fiore, and Baohua Wu. *The Design of Synchronization Mechanisms for Peer-to-Peer Massively Multiplayer Games.* Department of Computer and Information Science, University of Pennsylvania, 2004.

[3] Philipp Svoboda and Markus Rupp. *Online gaming models for wireless networks.* Institut für Nachrichten- und Hochfrequenztechnik, TU-Wien, February 2005.

[4] Jouni Smed, Timo Kaukoranta and Harri Hakonen. *A Review on Networking and Multiplayer Computer Games.* Turku Centre for Computer Science, April 2002.

[5] Björn Knutsson, Honghui Lu, Wei Xu, and Bryan Hopkins. *Peer-to-Peer Support for Massively Multiplayer Games*. Departure of Computer and Information Science, University of Pennsylvania, 2004.

[6] Seventh District's Distance Learning Center of Excellence. *Dead Reckoning- Coastal Piloting for Boat Crew and AuxNav.* http://www.auxetrain.org/Nav1.html , 2006-07-26.

[7] Nick Caldwell. *Defeating Lag with Cubic Splines.* GameDev.net, http://www.gamedev.net/reference/articles/article914.asp, 2006-07-26.

[8] Margaret DeLap, Björn Knutsson, Honghui Lu, Oleg Sokolsky, Usa Sammapun, Insup Lee, and Christos Tsarouchis. *Is Runtime Verification Applicable to Cheat Detection?* Department of Computer and Information Science, University of Pennsylvania, 2004.

[9] Wentong Cai, Francis B.S. Lee and L. Chen. *An Auto-adaptive Dead Reckoning Algorithm for Distributed Interactive Simulation.* School of Applied Science, Nanyang Technological University.

[10] Book of Hook. *The Quake3 Networking Model.* http://trac.bookofhook.com/bookofhook/trac.cgi/wiki/Quake3Networking, 2006-07-27.

[11] Book of Hook. *Introduction to Multiplayer Game Programming.* http://trac.bookofhook.com/bookofhook/trac.cgi/wiki/IntroductionToMultiplayerGameProgramming, 2006-07-27.

[12] Dave Weinstein, et al. *Multiplayer Tricks of Trade.* GDC 2004, San Jose, California.

[13] Zsolt Kenesi, Gábor Kiss, János Levendovszky and Sándor Molnár. *Optimizeing multiplayer gaming protocols for heterogeneous environment.* Budapest University of Technology and Economics, 2006.

[14] Marcel Busse, Bernd Lamparter, Martin Mauve and Wolfgang Effelsberg. *Lightweight QoS-Support for Networked Mobile Gaming*. SIGCOMM'04 Workshops, August / September 2004.

[15] Wu-chang Feng, Francis Chang, Wu-chi Feng, and Jonathan Walpole. *Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server.* OGI School of Science and Engineering at OHSU, 2002.

[16] Tom Jehaes, Danny De Vleeschauwer, Toon Coppens, Bart Van Doorselaer, Eva Deckers, W. Naudts, K. Spruyt, and R. Smets. *Access network delay in networked games.* NetGames '03, May 22-23, 2003.

[17] Johannes Färber. *Network Game Traffic Modelling.* Inst. Of Communication Networks and Computer Engineering, University of Stuttgart, 2002.

[18] Xu Chen. *Performance Analysis of Wireless Multiplayer Games on Terraplay Systems.* KTH, School of Information and Communication Technology, Stockholm, October 2005.

[19] Wikipedia.org. *PlayStation Portable.* http://en.wikipedia.org/wiki/PlayStation_Portable, Last modified 06:40, 31 July 2006.

[20] Wikipedia.org. *Nintendo DS Lite.* http://en.wikipedia.org/wiki/Nintendo_DS_Lite, Last modified 06:32 31 July 2006.

[21] Jesse Aronson. *Dead Reckoning: Latency Hiding for Networked Games.* Gamasutra, September 19, 1997.

[22] James Crawshaw. *Online Gaming: Invasion of the MMORPGs*. Light reading insider, VOL. 6, NO. 1, January 2006.

[23] Matthias Dick, Oliver Wellnitz, and Lars Wolf. *Analysis of Factors Affecting Players' Performance and Perception in Multiplayer Games.* Institut für Betriebssysteme und Rechnerverbund, Technische Universität Braunschweig.

[24] Harlan T. Beverly. *Online Gaming Lag: Definition and Causes with Case Studies*. Bigfoot Networks, Inc., 2005.

[25] Steve Jones. *Let the Games Begin: Gaming Technology and Entertainment among College Students.* Pew Internet & American Life Project, July 6, 2003.

[26] Entertainment software association. *2006 Essential facts about the computer and video game industry.* May 10, 2006.

[27] Major Gaming League. http://www.mlgpro.com/

[28] Cyberathlete Professional League. http://www.thecpl.com/

[29] Phil Taylor. *North American Cellular Data Forecasts (2005-2010).* Strategic Analytics, http://www.strategyanalytics.com. August 2005.

[30] Nitesh Patel. *Western European Cellular Data Forecasts (2005-2010).* Strategic Analytics, http://www.strategyanalytics.com. September 2005.

[31] Tanja Lang, Philip Bransh, and Grenville Armitage. *A Synthetic Traffic Model for Quake3*. Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia. 2004.

[32] Valve Software. http://www.valvesoftware.com/

[33] Steam. http://steampowered.com/

[34] Yanli Suo-Saunders. *Evolution of Mobile Handsets to 2011 and Beyond: market analysis and forecasts.* Analysis Research Limited, 2006.

[35] PunkBuster, Even Balance. http://www.punkbuster.com/

[36] Joel Zetterström. *A Legal Analysis of Cheating in Online Multiplayer Games.* Master Thesis, Department of Law, School of Economics and Commercial Law, Göteborg University, March 2005.

[37] Sudgir Aggarwal, Hemant Banavar, Sarit Mukherjee, and Samparh Rangarajan. *Fairness in Dead-Reckoning based Distributed Multi-Player Games.* NetGames '05, October 10-11, 2005, Hawthorne, New York, USA.

[38] Sebastian Zander, David Kennedy, and Grenville Armitage. *Dissecting Server-Discovery Traffic Patterns Generated By Multiplayer First Person Shooter Games.* NetGames '05, October 10-11, 2005, Hawthorne, New York, USA.

[39] Amjad Akkawi, Sibylle Schaller, Oliver Wellnitz, and Lars Wolf. *A Mobile Gaming Platform for the IMS.* NetGames '04, August 30 – September 3, 2004, Portland, Oregon, USA.

[40] Lothar Pantel and Lars C. Wolf. *On the Impact of Delay on Real-Time Multiplayer Games.* NOSDAV '02, May 12-14, 2002, Miami, Florida, USA.

[41] The Network Simulator – ns-2. http://www.isi.edu/nsnam/ns/.

[42] Sunwoong Choi, Kihong Park, and Chong-kwon Kim. *On the Performance Characteristics of WLANs: Revisited.* SIGMETRICS '05, June 6-10, 2005, Banff, Alberta, Canada.

[43] Mark Claypool. *On the 802.11 Turbulence of Nintendo DS and Sony PSP Hand-held Network Games.* NetGames '05.

[44] Xiaoxin Wang. *3G HSDPA Performance In Mobile Internet Connections.* Master of Science Thesis, KTH Microelectronics and Information Technology, Stockholm, Sweden, 2004.

[45] International Telecommunication Union. *Subjective audiovisual quality assessment methods for multimedia applications*. Telecommunication standardization sector of ITU, 1998.

[46] Bredbandstestet TPTEST http://www.tptest.se.

[47] TweakGuides.com. *Quake 4 Tweak Guide [page 8] Advanced Tweaking (Pt.2).* http://www.tweakguides.com/Quake4_8.html. Retrieved 2006-10-30.

[48] Tomb Raider: Legend. http://www.tombraider.com/

[49] Max Payne. http://www.maxpayne.com. Rockstar Games, Inc. 2003.

[50] Ground Theft Auto. http://www.grandtheftauto.com. Rockstar Games, Inc. 1997-2006.

[51] Need for speed. http://www.needforspeed.com. EA Games, 1994-2006.

[52] PricewaterhouseCoopers. *Global Entertainment and Media Outlook: 2006-2010 – Video Games.* P. 369-400, 2006.

[53] Massive Incorporated. http://www.massiveincorporated.com/.

[54] Daniel J., Van Hook, James O. Calvin, and Duncan C. Miller. *A Protocol Independent Compression Algorithm (PICA)*. MIT Lincoln Laboratory Project Memorandum, April 2, 1994.

[55] Sebastian Zander and Grenville Armitage. *A Traffic Model for the Xbox Game Halo 2.* Centre for Advanced Internet Architectures (CAIA), Swinburne University of Technology, Melbourne, Australia, June 2005.

[56] A. Markopoulou, F. Tobagi, and M. Karam. *Assessment of VoIP quality over Internet backbones.* In proceedings of IEEE INFOCOM 2002, New York, June 2002.

[57] Vtun – Virtual Tunnels over TCP/IP networks. http://vtund.sourceforge.net.

[58] World of Warcraft Community Site. http://www.worldofwarcraft.com.

[59] 2404 – Baattlefield 2 Review. http://www.2404.org/reviews/10/Battlefield-2-Review. Retrieved 2006-11-15.

[60] AGEIA PhysX by Ageia. http://www.ageia.com/.

[61] Tcpdump – dump traffic on a network. http://www.tcpdump.org/tcpdump_man.html. Retrieved 2006-11-30.

[62] Martin Heusse, Paul Starzetz, Franck Rousseau, Gilles Berger-Sabbatel, and Adrezej Duda. *Scheduling Time-sensitive Traffic on 802.11 Wireless LANs.* In proceedings of the 4th COST 263 International Workshop on Quality of Future Internet Services (QoFIS 2003), Stockholm, Sweden, October 1-3, 2003.

Appendix A
Game quality determination – survey form

The following form was used during the game quality surveys used in the thesis to get a figure about what performance acceptable real-time multiplayer game quality require.

The form is divided into two parts. The first part includes questions about the test persons gaming background with the main purpose to work as a weight when summarize the results. In the second part the test persons are presented with network configuration samples. Each sample is played for two minutes after which the test persons are asked to answer two questions and optionally give a comment. Both questions relate to how the game quality of the just played session was perceived. First the test person should mark his opinion about the quality with a cross on a ten centimeter wide line (left end of the line represent the worst quality while the right end represents the best quality). Following question is a yes/no question about if the quality was experienced as acceptable. These questions are repeated for each sample (20 samples used in each survey + 2 test samples). The survey took, all in all, about one hour to perform and due to limited number of game stations, a maximum of three test persons performed the survey at once.

Game quality survey

Initial questions

Put an X in the box or on the line where it best matches your answer. Comments are optional.

How often do you play video/computer games?			
[ ] Once per week	[ ] Several times per week	[ ] Once per day	[ ] Several times per day

How many of above occasions are played online (multiplayer)?				
[ ] All/almost all	[ ] More than half	[ ] Half	[ ] Less than half	[ ] None/almost none

What kind of games do you prefer when you play online?			
[ ] FPS	[ ] RTS	[ ] Sports	[ ] Racing
[ ] RPG	[ ] Other: ____		

Have you played Quake 4 before?	
[ ] Yes	[ ] No

Test samples

Sample A – No impairments		
Your opinion about the quality:	Worst Best	
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		

Sample B – With impairments		
Your opinion about the quality:	Worst Best	
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		

Sample 1		
Your opinion about the quality:	Worst	Best
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 2		
Your opinion about the quality:	Worst	Best
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 3		
Your opinion about the quality:	Worst	Best
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 4		
Your opinion about the quality:	Worst	Best
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		

Sample 5		
Your opinion about the quality:	Worst Best	
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 6		
Your opinion about the quality:	Worst Best	
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 7		
Your opinion about the quality:	Worst Best	
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 8		
Your opinion about the quality:	Worst Best	
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		

Sample 9		
Your opinion about the quality:	Worst Best	
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 10		
Your opinion about the quality:	Worst Best	
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 11		
Your opinion about the quality:	Worst Best	
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 12		
Your opinion about the quality:	Worst Best	
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		

Sample 13		
Your opinion about the quality:	Worst	Best
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 14		
Your opinion about the quality:	Worst	Best
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 15		
Your opinion about the quality:	Worst	Best
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 16		
Your opinion about the quality:	Worst	Best
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		

Sample 17		
Your opinion about the quality:	Worst Best	
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 18		
Your opinion about the quality:	Worst Best	
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 19		
Your opinion about the quality:	Worst Best	
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		
Sample 20		
Your opinion about the quality:	Worst Best	
Would you accept this quality for online play?	[ ] Yes	[ ] No
Comment:		

Thanks a lot for your participation!

COS/CCS 2006-17

www.kth.se