Stratified Protocols for Mobile Gaming

Sunny Koh

March 22, 2006

Acknowledgments

I have to say that this has been a very long process with many years in the making. Starting out with the goal to implement a Massive Multiplayer Online Role Playing Game on a Palm PDA became this look into how to work with the networking issues for any multiplayer mobile game. (And some dare say, even for non-mobiles as well)

I would like to first thank Associate Professor Ron Addie for having to endure working with me to complete this project. We have our disagreements over many issues but as it stands, I believe that both are happy that we are able to break new ground in the project.

Credit also has to go to Camelot Hobbies Private Limited for supporting my research project and allowing me to take time off instead of doing my job managing the technology and creative teams. I need to mention Thomas Lim which has worked with me through out the project giving me advice and pointing me to where the tasty points of research and so on.

To Anna Dobritt and Ting Chai Hing, thanks for your time discussing ideas and concept not necessary in direct relation to my project but just the same. Of course Bevis CyberRat Yeo for encouraging me to try out the Apple Mac Platform, without it, I would think I would be still trying to get Windows to work. Much thanks to Eugene O. Kolobov President of Stone Falcon, Russia for online discussions about computer Meta-gaming.

I also have to acknowledge the many volunteers which helped during the experiments in my research. I am unable to name them due to privacy requirements but they know who they are and thanks for all the help.

To my mentor Professor Andy Koronios, encouraging me to get into the Masters Program and inspiring me to contribute to society through research. We should get together more often like we use to.

And Finally and Most Important of all, many thanks and my eternal gratitude to My parents for supporting me all the way from young, letting me go overseas to study and to allow me to achieve my dreams in education.

Declaration

This report and the project documented by it were a requirement of Courses CSC8404 and CSC8405, Master of Computing Project A and Master of Computing Project B, Master of Computing, Department of Mathematics and Computing, University of Southern Queensland (USQ), Toowoomba, Australia.

A certify that the work reported in this thesis is entirely my own effort, except where otherwise acknowledged. This report is original and contains no material offered for the award of any other academic qualification at this or any other institution, or material previously published, except where due reference is made in the text.

Signature of Candidate
Student Number:D95300212

Date

Contents

List of Figures

Abstract

Mobile Games are becoming very popular. Players play games under various conditions and so we need protocols which can adapt. However, unlike the desktop, there are many issues with that of mobile communication. This dissertation proposes a *stratified protocol* to be implemented in order to solve these communication problems more effectively.

The protocol itself dynamically adapts to the changing bandwidth capacity of the network. The term stratified means that the payload in the protocol packets is subdivided into different categories, or strata, of data which are handled differently depending on network conditions.

Such a protocol needs to be *responsible* as it must make decisions which do not worsen the problem of network congestion when it arises.

An experiment was conducted with a view to determining whether it is important to address difficulties with technical quality of communication. This experiment showed that gaming satisfaction of players does depend on technical quality of communication and that the way in which this dependence on technical quality is exhibited depends upon the particular gaming genre. This dependence on genre and the fact that different games make use of communication services differently suggests that stratification of communication protocols could be useful.

Chapter 1

Introduction

Games have come a long way since MIT Lab's SpaceWar! in 1961 [Mar01]. Computer games have became a multi-billion dollar industry with game publishers pumping in millions of dollars per software title. [MP03]

Personal digital assistants (PDAs) & mobile phones are becoming a part of everyday life. The ease of use and the convenience makes it an indispensable tool for people on the go. Evolving from an address book, date planner and to do list or phone, the latest PDAs & mobile phones in the market offer processing capabilities that rival most first and second generation desktop computers. (The first generation of Nokia 9000 series communicator PDA/phone hyrid runs on 80386 platform with NEOS-DOS as the operating system.) Some run at an amazing 800MHz in a RISC format which is comparable to desktops today. From a purely grayscale LCD and touchpad, today we see PDAs & mobile phones with amazing 16 million color displays, vast amounts of storage and expandability (With more than 4 GB of data with the addition of the compact flash thumb drive same as the one found in the iPod Mini.) Applications are made for every conceivable field including entertainment.

Entertainment on a PDA & mobile phones can be described as following the progress of the computer gaming scene. From the crude puzzle games to the current graphical adventures. Simple games like card games have now given way to Sim City Classic for the Palm Pilot (A full copy of priginal Sim City game) and sprint based RPG games. As proof of a growing market, we see big players in the entertainment industry like Sony launching a handheld RPG product titled after their popular Everquest

Figure 1.1: Nokia N-Gage QD (Second Generation Nokia Phone/Mobile Console Hydrid)

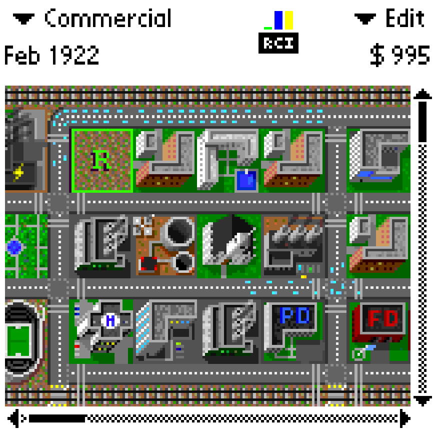


Figure 1.2: Simcity Class for the Palm

series. With more developers joining the bandwagon, we can see enhancements and development in these types of games.

The next generation of mobile devices will include peer to peer multiplayer capibilites through technologies like Bluetooth and WiFi. Devices like The game Boy DS boast peer to peer play with up to 16 players and even the current Tapwave Zodiac supports 8 player via Bluetooth. Clearly there is a reach into the multiplayer mode.

With the expansion into these areas, it can be said that overcoming the network protocol issues with wireless technology will be a major factor. There is a need for an adaptive protocol as there are highly variable operating conditions in mobile gaming. As these are the early days of mobile multiplayer gaming, it is important to find the ideal solution to solve these problems. The quality of communications at the moment is patchy at best. This dissertation hopes to address some of the issues in this operating environment.

In Chapter 2, we survey the background of the mobile computing environment in general, including considerations of mobile computing and the technologies supporting this computing environment. In Chapter 3, we discuss the history and background of game communication, including the methods and protocols used and what is communicated in order to support multiplayer gaming.

In Chapter 4, a brief introduction to middleware is given, including a discussion of what gaming middleware in general and some analysis on current network middleware systems being offered today. In Chapter 5, based on the understanding gained from the previous chapters, a set of requirements are drawn up for mobile gaming. Technical, social and commercial aspects are discussed and why they should be given consideration.

In Chapter 6, stratification is proposed and examples on how stratification could work to improve common mobile gaming scenarios. By addressing the requirements set out in Chapter 5, the communication stratification concept is explored. In Chapter 7, a proposed design of middleware providing stratified communication for gaming

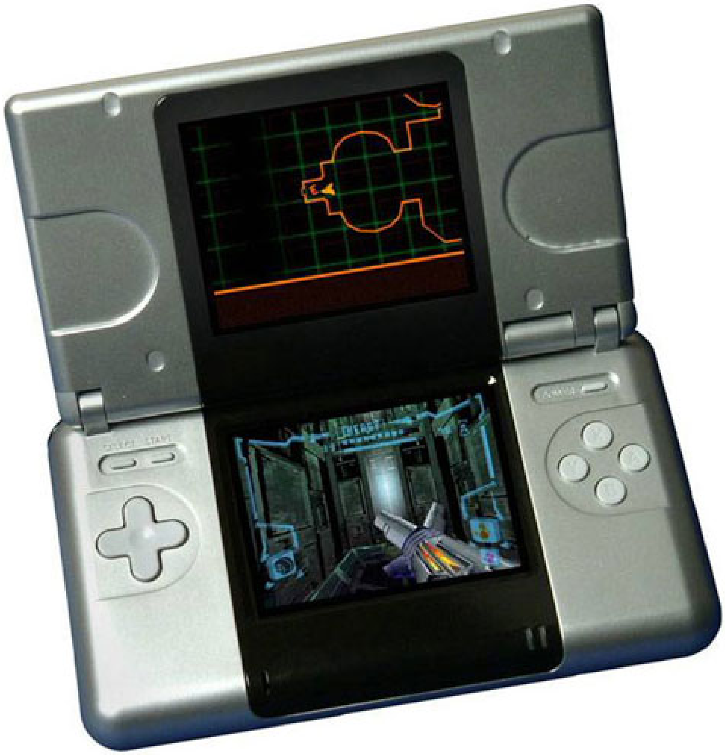


Figure 1.3: Nintendo Gameboy DS

software is presented.

Chapter 8 describes a series of experiments in which games are played over a communication network with deliberately impaired technical quality and the results of these experiments are analysed. A series of hypotheses concerning the response of players to poor technical quality of communication are tested.

Concluding remarks and discussion of further work is presented in Chapter 9.

Chapter 2

Existing and future Mobile Games

In this chapter, we would examine the devices that people play mobile games with, the technologies for having multiplayer games, some popular games and the methods to create them.

2.1 Introduction to Mobile Games

What is a mobile game? Why are people focusing on them? Why is it “different”?

Quite simply, a mobile game is a game which can be played on a reasonably portable device. Many would argue that this is a rather new market of game development but in effect, it has been around for some time now. Many can remember the simple electronic toys that have only a single game which you can carry from place to place. These devices form the beginnings of the mobile games industry.

As the PC and console markets become more competitive, more and more money is spent by the big players to attract the customer base. As it is, it costs more then US $10 Million to produce a Massively Multiplayer Online Game (MMOG). [MP03] Small players are being muscled out of the market as they don’t have the budgets to keep up with these companies. However as the demand for “mobile” and portable games grow, these small companies are finding themselves a niche to be filled. Not only that, we are beginning to see major players advance the mobile gaming platform with new technologies. [SC04]

The issue about mobile gaming is that due to the platform they are to be played on, there are real design limitations. Mobile gaming platforms are designed compromised with one feature is traded off for another for the sake of mobility. Not only the platform has these issues, but the communication technology too has similar issues.

2.2 Mobile Gaming Devices

To start on the discussion existing mobile games, we should first look at existing mobile gaming devices. We limit this discussion to devices which can support multiplayer

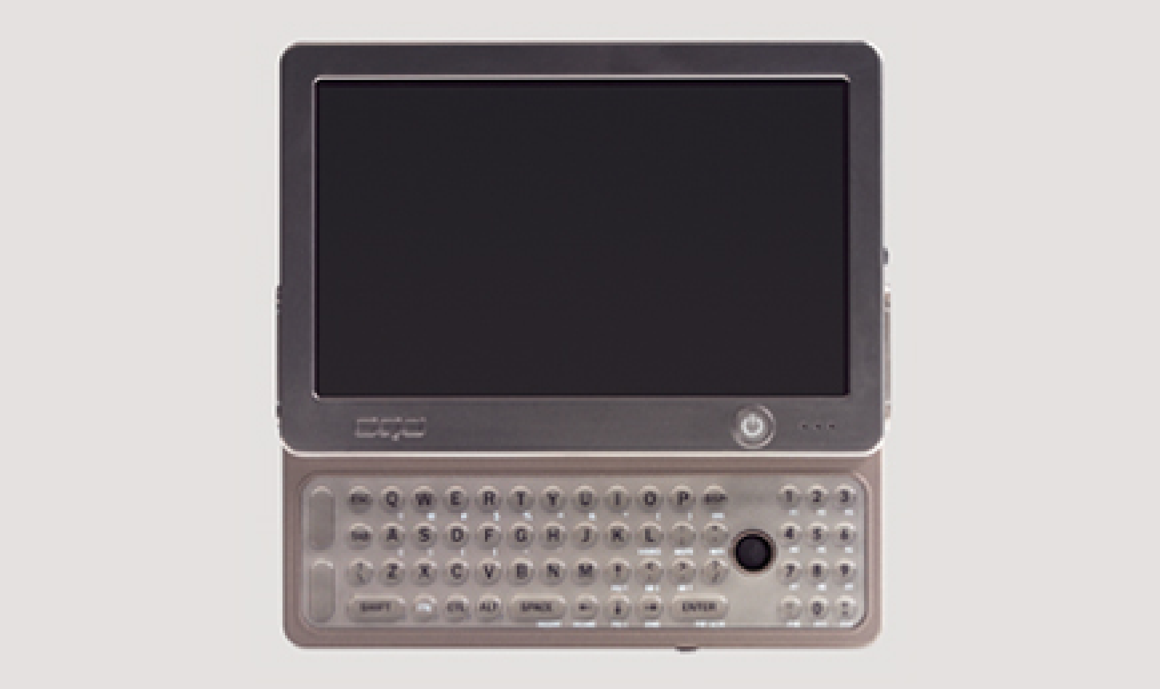


Figure 2.1: OQO

games as opposed to single player or virtual multiplayer.

2.2.1 Mobile Consoles

The gaming console is considered a little brother to those machines that you connect to your television. Such devices allow players to “game on the go”. Almost all devices in this heading are purely for entertainment.

Popular current devices include The Gameboy (Figure 1.3 shows the next generation Gameboy) and others. Games are stored on cartridges with a single game per cartridge.

It should be noted that the next generation of mobile consoles will have wireless connectivity. For example, the Sony PSP (Figure 2.2) has both WiFi and an InfraRed communications port. This is to allow for more multiplayer options.

2.2.2 Notebook Computers

Notebooks and laptops don’t have the same limitations as most other mobile device but still are subject to the same communication issues faced by their less powerful cousins. (This is in regards to the wireless communications methods. Wired communication methods are similar to those of desktops.)

It should be noted that not all notebooks are created equal. The ultra-ultra lights like the OQO (Figure 2.1) can fit into a pocket. While others like the Alienware 51m

Figure 2.2: Sony's PSP

are so heavy that it is a “portable” desktop computer. (These systems are designed specifically for gaming and use Desktop components for the extra performance at the cost of almost no battery life and weight.)

2.2.3 Tablet Computers

Tablet computers can be defined as a notebook with a writable screen. These system differ slightly from laptops as in order to use them you don't need to use a keyboard. Instead similar, to a PDA, you can simulate key presses using an on-screen keyboard. Some tablets can be converted to notebook mode by opening the system and twisting the screen 180 degrees. The communication profile for this class of devices is the same as the notebook computer.

More commonly found in the enterprise market, tablet computers are slowly filtering down the line. Many educational institutions are or have considered Tablets as a suitable replacement for the textbooks that students carry from place to place. With that in mind, it should be noted that this is largely a untapped market with no games specifically designed to make use of the tablet's unique features.

2.2.4 PDA

Personal digital assistants (PDAs) are handheld computers which replace the diary or day planner. These devices have progressed to the point where they are internally a mobile computer which is handheld. PDAs are expandable platforms in which 3rd party programs can be written, thus forming the basis for the PDA gaming industry.

Figure 2.3: Alienware 51m

Figure 2.4: Fujitsu’s Tablet PC

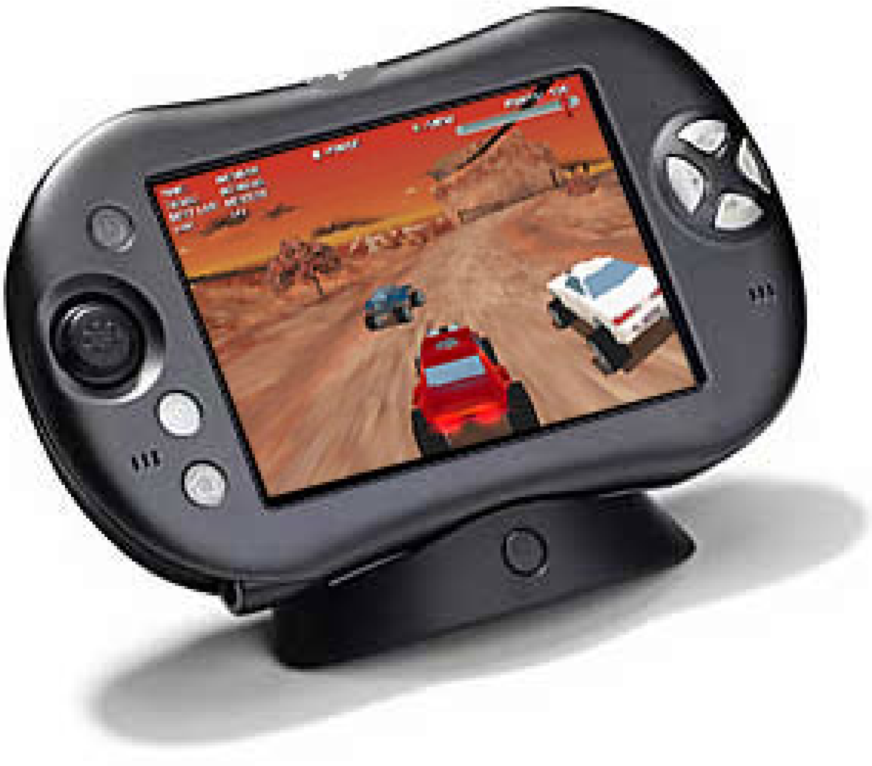


Figure 2.5: Tapwave's Zodiac

The PDA market is dominated by 2 operating systems, namely Palm OS by Palmsource and Windows CE by Microsoft. Palm has been in the market since 1996. [Myk00] It should be noted that the games written for the Palm started as a hobbyist industry and this is growing as more companies get involved. However, with the late arrival of Windows CE, major entertainment companies have been releasing titles for that platform like Sony's Everquest (Figure 2.9)

It is very important to note that as there are many different models of PDAs, there are many different modes of networking. (These connectivity options are sometimes more expensive than other devices of similar functionality due to size.)

The other issue to note is that almost all of these devices do not contain a GPU. And in order to keep the PDA power up longer, most don't have a fast processor either. However, Tapwave's Zodiac (Figure 2.5) is one of the exceptions where we can see a convergence device. This device has an onboard 3D GPU and is primary designed for gaming.

2.2.5 Mobile Phones

A mobile phone by definition is a telephone which uses wireless technology to place a phone call. These devices have become very common place and the rate of adoption is always increasing. With lots of people carrying mobile phones, manufacturers must find a new reason for people to upgrade their phones. In addition to upgrades for technologies, handset makers try to pack more functions into the phone.

Games on mobile phones can be played over the SMS system. (Very common as a tie in with another medium like radio.) Games on the mobile can be embedded into the firmware or in some sort of removable storage. With the possibility of wireless internet with WAP, GRPS or 3G, we can have multiplayer games over wide distances on the go.

The current crop of new mobile phones offer functions similar to that of a PDA. In fact, there is one segment of the market called the smart-phone, which is a PDA / Mobile hybrid. There are three operating systems of note, Microsoft smartphone, Palm OS & Symbian OS. These devices provide the same platform as that of a PDA but with phone functions. (And connectivity via the phone network.)

The smart-phone is not only a convergence device, Nokia has came out with the "second generation" of convergence phone: N-Gauge. (Figure 1.1) This is a mobile phone cum mobile game console which has functions like Bluetooth and an analog controller. With this device Nokia has successfully manage to get big name titles onto the platform.

For those who do not have access to such mobile phone handsets, the average new phone now supports JAVA as a in-built feature. (This is done using the Sun's J2ME standard) This allows teams of programmers to create simple puzzle type games on the platform. There is a word of caution as most handset makers even when complying with the J2ME are not compatible with each other.

Figure 2.6: Nokia's 7600

2.3 Multiplayer Games

The lure of a real thinking opponent is better then a computer's scripted Artificial Intelligence. This draw of competition is creating new industries from internet cafes which specialized purely in network gaming (Within a local area network, the response time between computers communicating is minimized) to service industries like subscription based massively multiplayer online games.

2.3.1 Types of Multiplayer Games

There are many classifications for multiplayer games. Here, the context to describe the different types is that of the number of human opponents. This maybe slightly different from convention, but by placing them in this context, one can identify the different issues faced by each one.

Peer to Peer

The peer to peer mode can be defined as playing a single opponent in this context. This is done traditionally using null-modem, modem or IPX networking in the early 1990s and before. (Though it is suggested elsewhere in this text that it can also refer to self contained clients talking to each other, but under this definition here it refers to that of a single opponent)

Network Play

Network play would refer to having 2 or more opponents in the same game. Starting with IPX networking in the past, we have seen it move to TCP/IP. Usually there is a host computer which other clients connect to which manages the synchronizing of the clients. This might or might not be a dedicated server.

Massively Multiplayer

The last type is that of the massively multiplayer. This can be define as having about 2000 to 3000 or more simultaneous clients connected to a server cluster in which all the players interact in the same virtual environment. [MP03] These clusters are mirrored to support many more users on the internet. One of the first MM game is Ultima Online. See Section A.1.4.

2.4 Current Product Offerings

In this section we survey some of the more popular and noteworthy mobile games which are in the market today. This is to allow us to gain an understanding of the direction in which the mobile gaming industry is heading.

Bejeweled is shown in Figure 2.7. In this game, you are to match a series of shapes by switching over 2 shapes next to each other. When 3 or more shapes match, those shapes which match up disappear and more shapes are added to the top to fill the screen. This represents the major portion of mobile games. Having simple repeating puzzles seems to be very popular in the market.

The next game we would want to consider is Warfare Inc. (Figure 2.8) There are 2 very important concepts demonstrated with this game. The first is that this is the first real time strategy game for the PDA platform. Players familiar with the gather, build, fight game play would feel right at home with this game. It is achieved in a very limited platform. The second item of note is that the game supports multiplayers. This is also another first. (With their version 2) players can connect using Bluetooth to fight each other.

Another very popular genre which made it to the PDA is that of the RPG. Sony has created a series based on their Everquest line of products as a single player game for the Pocket PC. (Figure 2.9) Players have the same functionality as in a full RPG on the computer with the same elements. It should be noted that the graphics used are not of the 3D type but of sprites.

The last game I would want to mention is Tibia ME. (Figure 2.10)This is a Multiplayer Online Role Playing Game for the mobile phone. Players now play this genre of games using their mobile phone in certain areas in europe. The graphics are very simple and so is the interface. Claimed by the developers as a MMORPG, this game is perhaps better classified as a Graphical MUD. [Pal03]

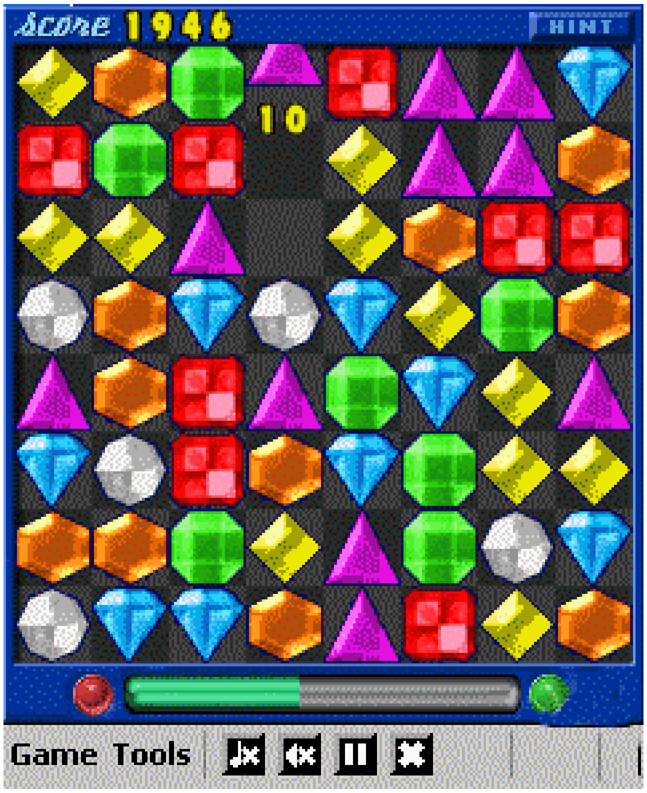


Figure 2.7: Screenshot from Bejeweled for PocketPC

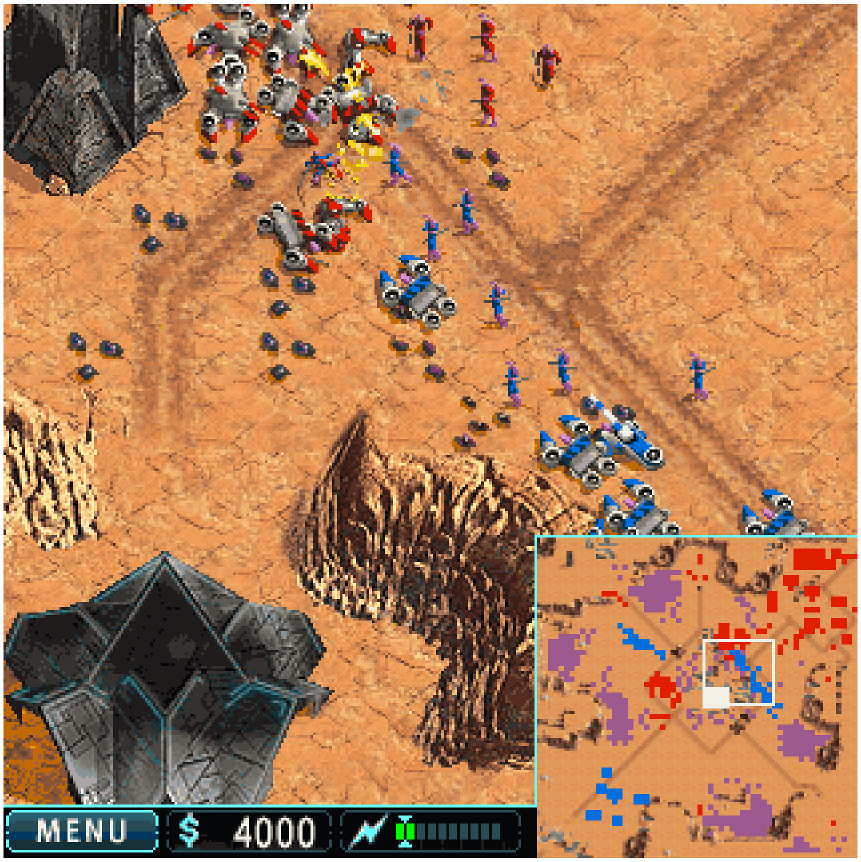


Figure 2.8: Screenshot from Warfare Inc for Palm OS

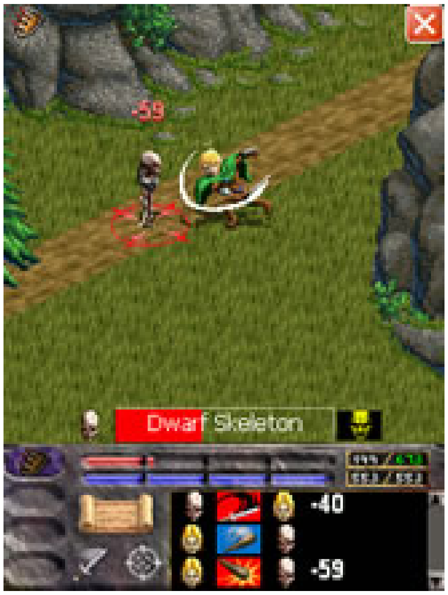
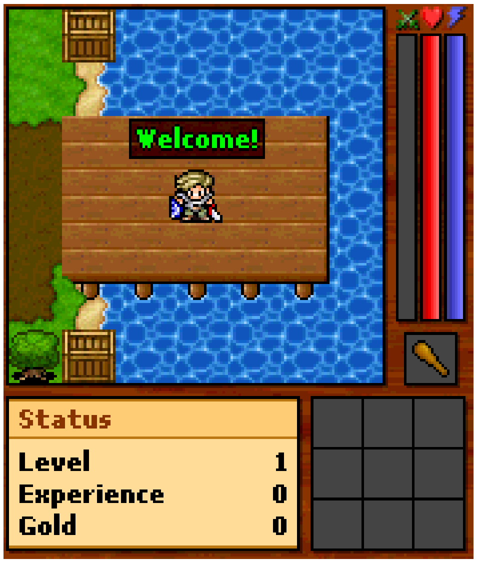


Figure 2.9: Screenshot from Everquest for Pocket PC



Status

Level 1
Experience 0
Gold 0

Figure 2.10: Screenshot from Tibia ME for Mobile Phone

2.5 Mobile Networking Technologies

There are many methods for connecting devices with each other or with a server. In this section we briefly go over the main technologies. It should be noted that the Push-Pull concept is very common for Mobile Phones while other devices like the PDA and notebooks use the client/server model.

InfraRed

Using the InfraRed spectrum of light, data can be exchanged between two devices. Used in the mobile platform as a fast method to exchange small packets of data like business cards, it has a very short range and needs line of sight between the 2 devices which want to communicate with each other. InfraRed is often used in 2 player games in the Palm OS. This only allowed for a turn based game in which a update is sent via the InfraRed link.

Wired

Wired would refer mainly to 2 types of connection. The first of which would be that of a physical modem. This is used by PDA to connect to a phone line for connectivity. The second is that of the network. In this case, the device is directly connected to the network. There is a third type use by some mobile game consoles in order to directly connect to each other.

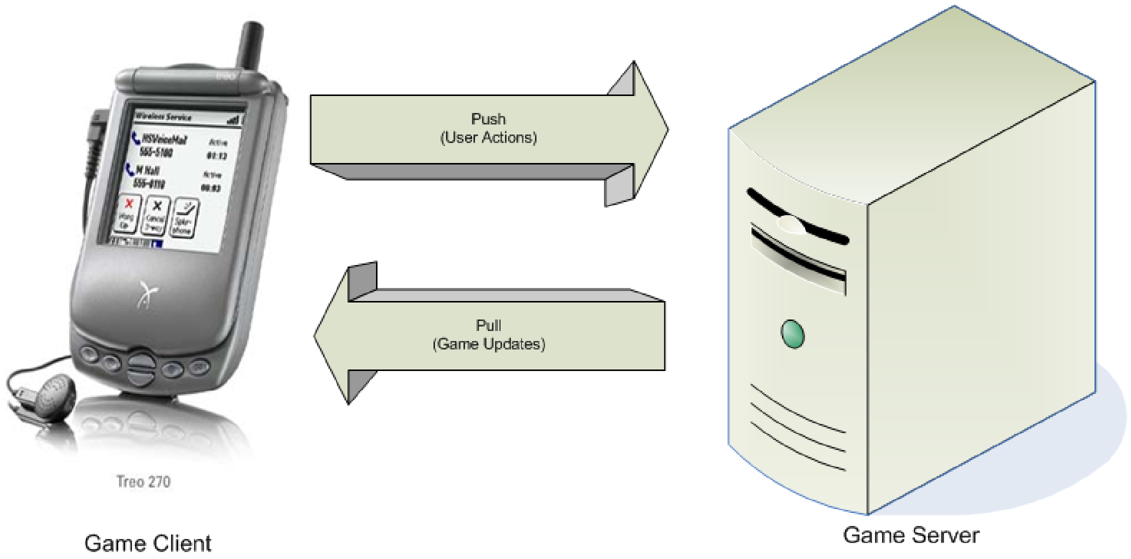


Figure 2.11: Mobile Phone Gaming Model

SMS/MMS

Short Message Service is a system in which mobile phones can exchange text messages with one another. Multimedia Message Service is graphical and multimedia version of SMS. These 2 protocols are not often used in player to player games but are common in wide scaled contests. Strictly speaking, these barely qualifiy as a game however, there are some systems out there that allow a game to be played over SMS.

Bluetooth

Bluetooth is a wire replacement technology for the creation of Personal Area Networks.(PAN) Common found in mobile phones, it is the communication method of choice to link the laptop or PDA to the mobile phone system. It should be noted that it has a very limited range depending on the rating of the device. However, we can see multiplayer games implemented using Bluetooth like Warfare Inc (Figure 2.8).

Mobile Phone Systems

This blanket term would refer to the digital protocols and systems that mobile phones use to link up with the telephone exchanges. As digital data is 1s and 0s, this connection technology can be use to carry pure data as well as voice. Of all the technologies this is the one with the longest range. Currently mobile phones cannot support data rates as high as a 56K modem. With the introduction of 3G, mobile phones will be able to support higher data rates in the future.

WAP

Wireless Access Protocol can be said to be the equal of HTTP in the web browsing arena. WAP uses similar techniques to *get* and *post* data over wireless link taking into account the difference in the network. Games similar to those using the web browser can easily be implemented using WAP [Sep01]. WAP is, of course, not actually a communication medium. WAP relies on other protocols used by the mobile phone systems to provide the communications link.

WiFi

Wireless Fidelity is another name for the 802.11 family of protocols to provide ethernet-like technologies over the wireless link. This very popular technology is enjoyed by mainly laptop users and PDAs. (It should be noted that the PalmOne Palm OS PDAs are always trailing behind of their Sony counterparts due to power issues.) Users can connect to the Internet through “Hotspots” in which WiFi networking is provided. WiFi can support fairly high data rates, up to 54 Megabits per second, although this varies depending upon the environment conditions.

2.6 Mobile Software Development

The profitable development of mobile games is difficult as best. This is because of the wide number of platforms each with a slightly different implementation of the same standard, yet different enough that a port for each platform is created. This does not make business sense as it is a very time consuming application [Bik03].

On top of the lack of a common platform standard across the board for mobile phones, there is a mess of developing options include items like BREW or J2ME and so on [Pal03]. The common midlet standard can be used in the development of mobile software but it is slow and unable to make use some optional advanced equipment like graphical processor and the like. [Fox01]

On the Palm OS platform we don’t have many of the these problems as C is the language of choice for the development of Palm software. With a well establish API with numerous books and online references, it is not a surprise that they are many third party developers for this platform. Even with Tapwave’s addition of a gaming engine, developers can create software applications that would run on almost any Palm OS Device and use the gaming engine on the Zodiac if there is one.

The story is similar for the Microsoft environment. The Visual series of IDE provide a consistent and unified approach to development of software titles. As Visual C and Visual Basic are the primary development platforms for Windows CE, applications can be ported from the desktop environment to the PDA quite readily.

The last platform that we should draw our attention to is that of mobile devices using linux. These may form a small percentage of devices now, but it is growing as an alternative to the big 3 OSes in the market place. Development for this platform is exactly the same as that for the desktop version of linux.

The transfer of software for debugging and distribution depends on the platform developed for. This can be as simple as a cable to link the device to the computer or using some wireless links.

2.7 Historical Comparison between old PC games and Current Mobile Games

Trends are an important part of any industry. Looking for such trends before it becomes mainstream can be the difference between being a market leader or a market follower. Short of a crystal ball, one must rely on the analysis of the information at hand.

It very interesting to note the progress of PC games over the years. From simple games which run under DOS to the impressive titles that tax even the fastest gaming rig, one can see the evolution in hardware and software development as games push the envelope.

And yet, what is the progression path for mobile games? Can we predict what is going to happen? Well, we can make an educated guess in what is going to happen. As the trend of the development of PC games and mobile games are very similar in terms of stages.

How is this so? Well, lets start with the Palm OS. Programming for the Palm seems a throw back the the good (or was that bad) old days of DOS programing. This is a time in which you would have to do most of the coding by hand and implement the complex functionality that Windows now provide. With this set of tools, we notice that games create have a striking resemblance with their ancient DOS counterparts. Simple games, not so much in graphical details and the like,

Of course the style of the game does not really underline the inner workings. Again with the Palm train of thought, we see that multiplayer games started with the use of the Infrared port. Wait a second, InfraRed port? A serial device in which data can be send and received. That would remind us about modems and games which require modems to play each other.

Okay, since we have established that the past of both platforms are similar, how about the programming techniques? Sprites anyone? In the past when there is no GPU on the PC platform, most graphics on screen are sprites, that is, a shape in graphical memory that can simulate movement by fast switching. Some of these sprites give the appearance of 3D using their detail. This was before the first true 3D system came out. Sprites are now being used in mobile gaming platforms again as having the same limitations as when game development was in those old days.

The GPU is the final piece of the puzzle. Most computers did not have a GPU unit in the early days of the desktop. (The term GPU was coined much later) And when it came it, it came out slowly. Notice that most mobile devices don't have a dedicated graphics unit now but now new models are coming out with one.

From the argument above, we can see that we are repeating history again, but this time in a different platform. But since we know where we are going, we can leap frog forward. Lessons learned and technological improvements should rapidly apply across the board. The key is to think beyond what we did before and to do the smart thing earlier.

Chapter 3

Gaming Communications

In order to include multi player capabilities in a game, there are always a few options. The simple one that has been effective in the past is that of having players take turns on the same computer or having a split screen view.

With the advancement of technology, such methods are dated as players demand real time interfaces in games that they play. As such Communications between different instances of games has become an important issue to be discussed and researched.

3.1 Historical Game Communication Methods

Short of having both players connected to the same computer, there are many ways in which games connect to each other. This section discuss some of the methods used historically.

Peer Client

A Peer Client can be described as a computer running the same game software in every instance where the game is being played. Typically, one client would become the host to which all the other clients are connected.

We begin first with the modem link (Figure 3.1). This is when one client calls another machine to establish a 2 player environment where each player uses his or her own machine to play the game. The state update of the game is sent through the modem. Some games also allow users to establish a modem hookup without even connecting to the POTS (Plain Old Telephone System) through a phone wire. This still remains as an option in games today.

The second method is to use a null-modem cable. This simple cable is shown in Figure 3.2 connects 2 nearby computers using the serial port of both. What happens is that the transmit and receive wires are crossed so on one end it is transmit and the other receive. This however has a limitation in terms of length as the longer the cable,

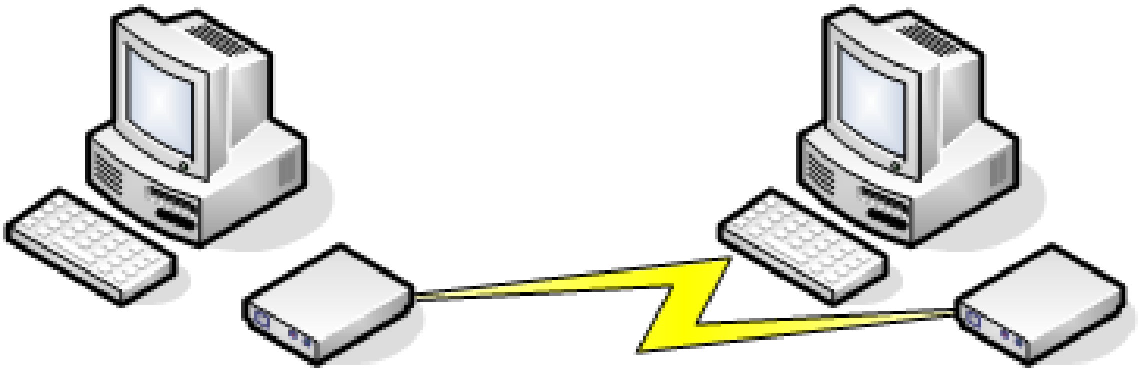


Figure 3.1: Modem Connection

the less reliable it is. Still, it is possible to connect 2 computers up to 100 meters apart using this method.

It can be noted that this method can only connect 2 computers together to provide multiplayer service in a peer to peer model. However, Blizzard Entertainment in Diablo allows 3 computers to connect through cabling, this is done using a mixture of a null-modem cable and a lap-link parallel cable. What if you need to connect more players? You would then need a network.

A Computer Network is a series of linked computers using a common set of protocols. The most popular standard used to be the Novell Network IPX protocol family. This was used by many games until TCP/IP became popular dominant. The details of this protocol are discussed in the next section.

Server based Gaming

Discussions about multiplayer games often concentrate on the peer client model. It is also worth to mentioning Server based Gaming. This form of games existed earlier than the above mentioned games. [Kos02] These are games which are stored on a server, and accessible through generic terminals.

Included in this category are Multi-User Dungeons (MUDs) and other forms of games. An interesting point to make is that not only people who have access to big mainframe type computers can play such games but the rise of the online service like Compuserve (Now part of AOL) and BBSes (Bulletin Board Services) provided access to the everyday masses.

In such services, people typically use a modem and terminal emulation software to dial up and connect to them. They then select from an option and they are placed into these games. These services may provide a single player or multiplayer environment depending on the game. These dialup services were very popular before the rise of the Internet as we know it today. In fact, cyberspace can be said to have evolved from such services.

A worthy mention is that of Baron Realms Elite. (Figure 3.3) This is a BBS Door

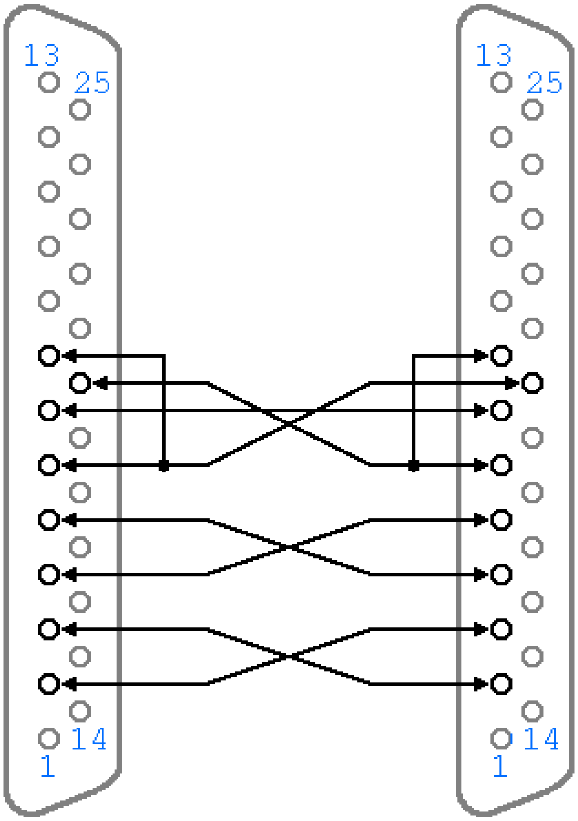
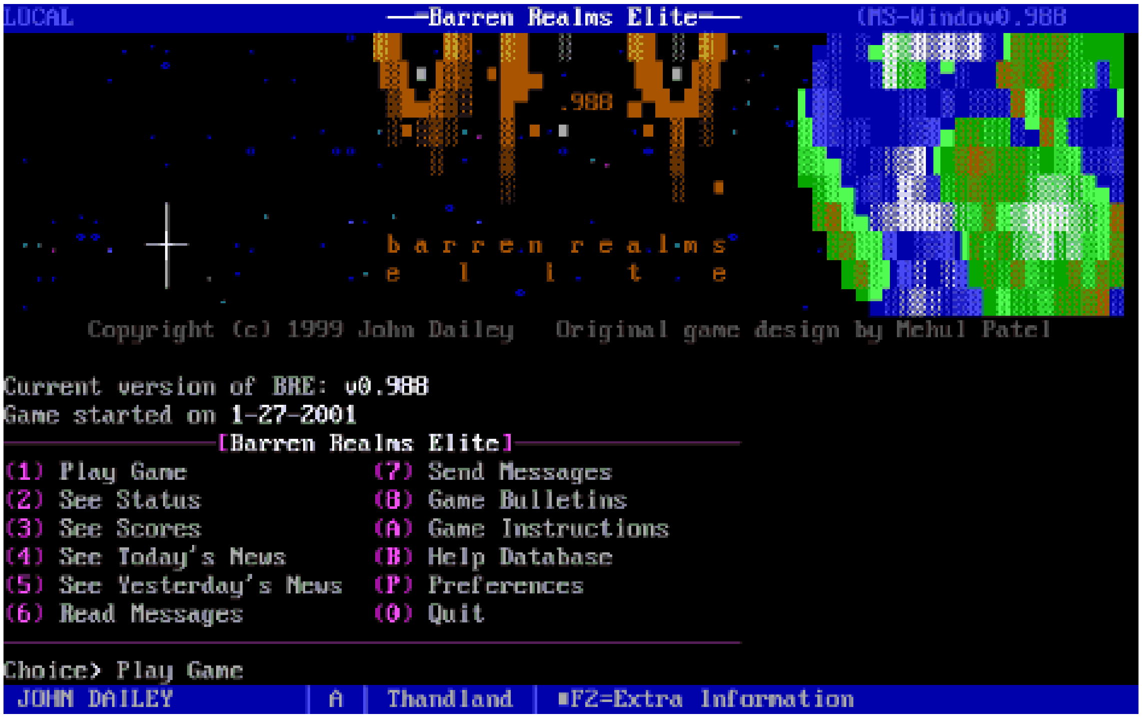


Figure 3.2: Null-Modem 25 pin Wiring Diagram



```
LOCAL                    ──Barren Realms Elite──              (MS-Window0.988

                              barren realms
                               e  l  i  t  e

      Copyright (c) 1999 John Dailey   Original game design by Mehul Patel

Current version of BRE: v0.988
Game started on 1-27-2001
────────────────[Barren Realms Elite]────────────────
(1) Play Game                  (7) Send Messages
(2) See Status                 (8) Game Bulletins
(3) See Scores                 (A) Game Instructions
(4) See Today's News           (B) Help Database
(5) See Yesterday's News       (P) Preferences
(6) Read Messages              (0) Quit
─────────────────────────────────────────────────────

Choice> Play Game
 JOHN DAILEY          | A | Thandland | ■F2=Extra Information
```

Figure 3.3: Screenshot from Baron Realms Elite

Game. (Door Games are standalone software that the BBS software runs as in a shell.) Which allows multiplayer not only in a single Bulletin Board but allows multiple Bulletin Boards to link up via the Fidonet Mail Exchange System to allow “fighting” across systems.

3.2 IPX

Created by Novell in the 1980s as a protocol for provision of Networking Services across clients and server through their popular NOS Novell Netware, [Cis03] IPX is the protocol of choice to support more then 2 players. As the most common networking protocol used in DOS, most multiplayer games support it.

One of the issues with IPX at least during the DOS era was the difficultly of setup. As Games are memory intensive software, it is hard to balance the requirements to load the different layers and the lower memory requirements. With the advance of Windows 95, the memory is handled by the operating system.

With the rise of the publicly available Internet, IPX is slowly phrased out from Games in favor of TCP/IP. This is because TCP/IP can support bigger Wide Area Networks and is the de-facto standard protocol. (Even Novell has decided to support TCP/IP in their Netware Line since 1995) It can be safely assumed that any widely installed protocol would be what developers would be using to support their network

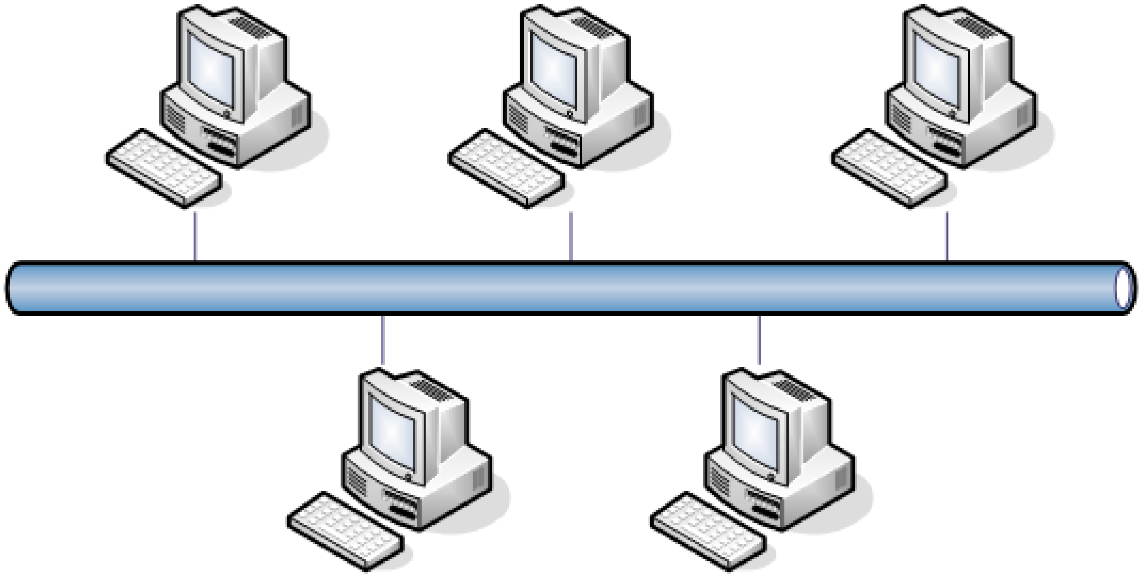


Figure 3.4: Ethernet

play infrastructure.

3.3 Brief overview in TCP/IP and UDP

With his paper “Information Flow in Large Communication Nets” (July, 1961), Leonard Kleinrock proposed the idea of packet switching which later developed into the TCP/IP Suit of protocols and subsequently laid the foundations of the Internet Today. [Kle04]

Packet Switching is a computer network protocol that breaks one big message into very small packets and then sends them through a forwarding network and the destination reassembles the messages. The DARPA network was designed as a network which can survive an attack which can take out some parts of the network.

The TCP/IP Suit has 4 layers which correspond to the 7 layer Open System Interconnect Model. These are the Link Layer, which happens the physical and electrical link between nodes, the network layer which handle basic communications through IP (Internet Protocol), the Transport layer which uses mainly TCP (Transmission Control Protocol) and UDP (Unreliable Datagram Protocol) and finally the application layer. See Figure 3.6 for more details.

In his book TCP/IP ILLUSTRATED, VOLUME 1, THE PROTOCOLS, Stevens defined that each layer of the TCP/IP Network Protocol has headers. [Ste94] Starting with IP, this header serves as the routing layer of the TCP/IP suit, it is used by routers to forward the packets from one part of the network to another. The header can be defined in figure 3.7.

The Next Protocol that we are interested in is TCP, which is a stream based protocol to connect 2 points. It provides recovery from packet loss and flow control. Flow

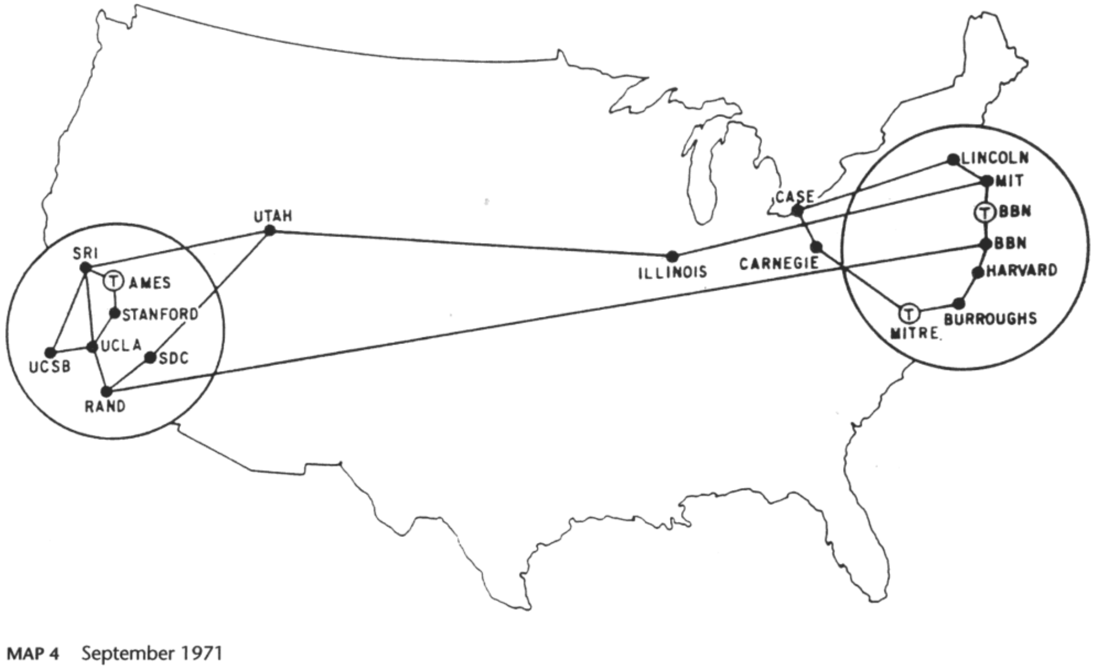


Figure 3.5: ARPA Network 1971 from “Casting the Net”, page 64

control can be defined as sending as many packets as the network and the source & destination can handle. This is dependent on the available capacity in the network. Common uses of TCP include the application Telnet, FTP or File Transfer Protocol, HTTP (for the World Wide Web) and SMTP (for mail). The TCP Header is defined in Figure 3.8.

UDP is a datagram protocol which is message based. It is unreliable as packets can be discarded for any reason, so there is no guarantee that packets will arrive. Despite this flaw, UDP is fast and is commonly use in applications like video and audio streaming. The UDP Header is defined in Figure 3.9.

3.4 Current Game Communications Models

Most Game Communications use either the client/server model the peer to peer model of communication. However, some games use both approaches at the same time.

Client/Server Model

The main type of communication model used in games is the client server model. (Figure 3.10) This is when all the clients pass the data onto the server for processing and redistribution.

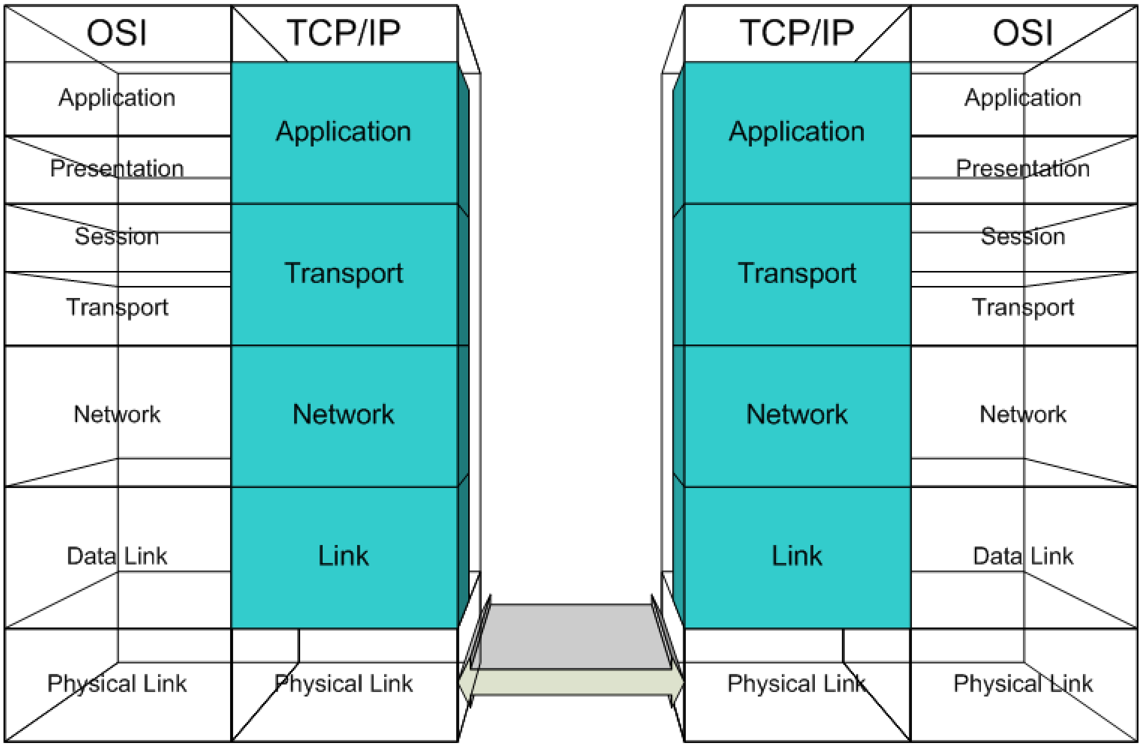


Figure 3.6: Open System Interconnect verse TCP/IP Layers

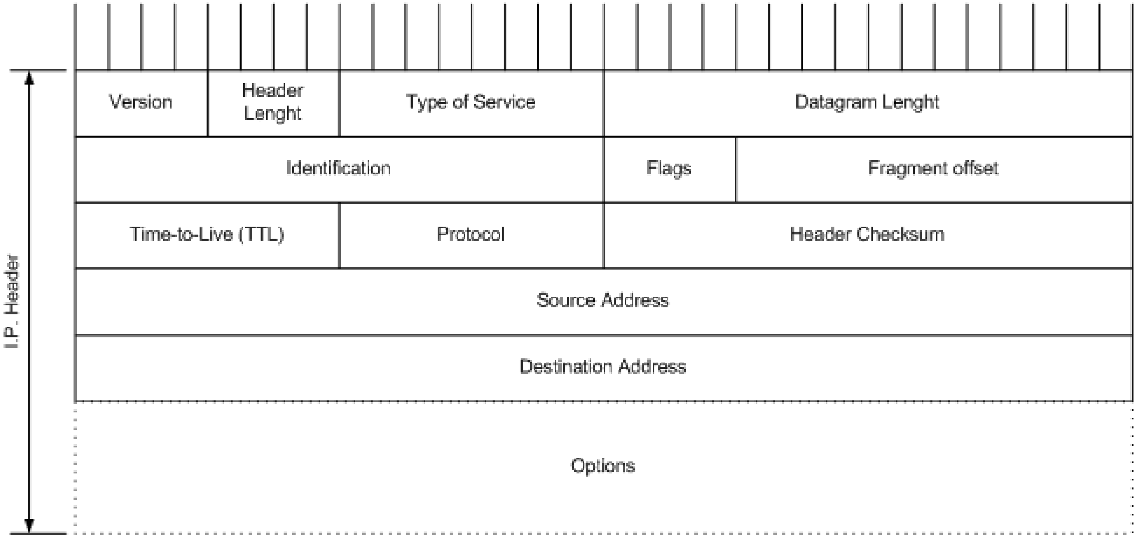


Figure 3.7: I.P. Header

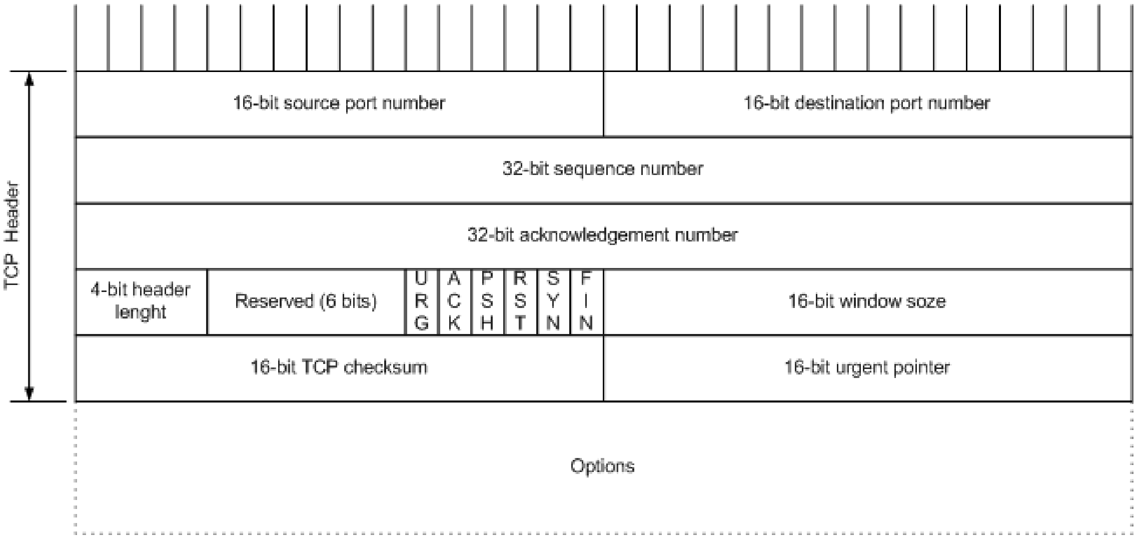


Figure 3.8: TCP Header

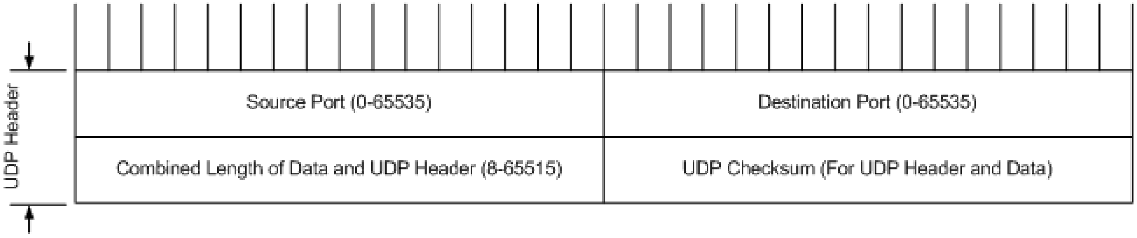

UDP Header
Source Port (0-65535)
Destination Port (0-65535)
Combined Length of Data and UDP Header (8-65515)
UDP Checksum (For UDP Header and Data)

Figure 3.9: UDP Header

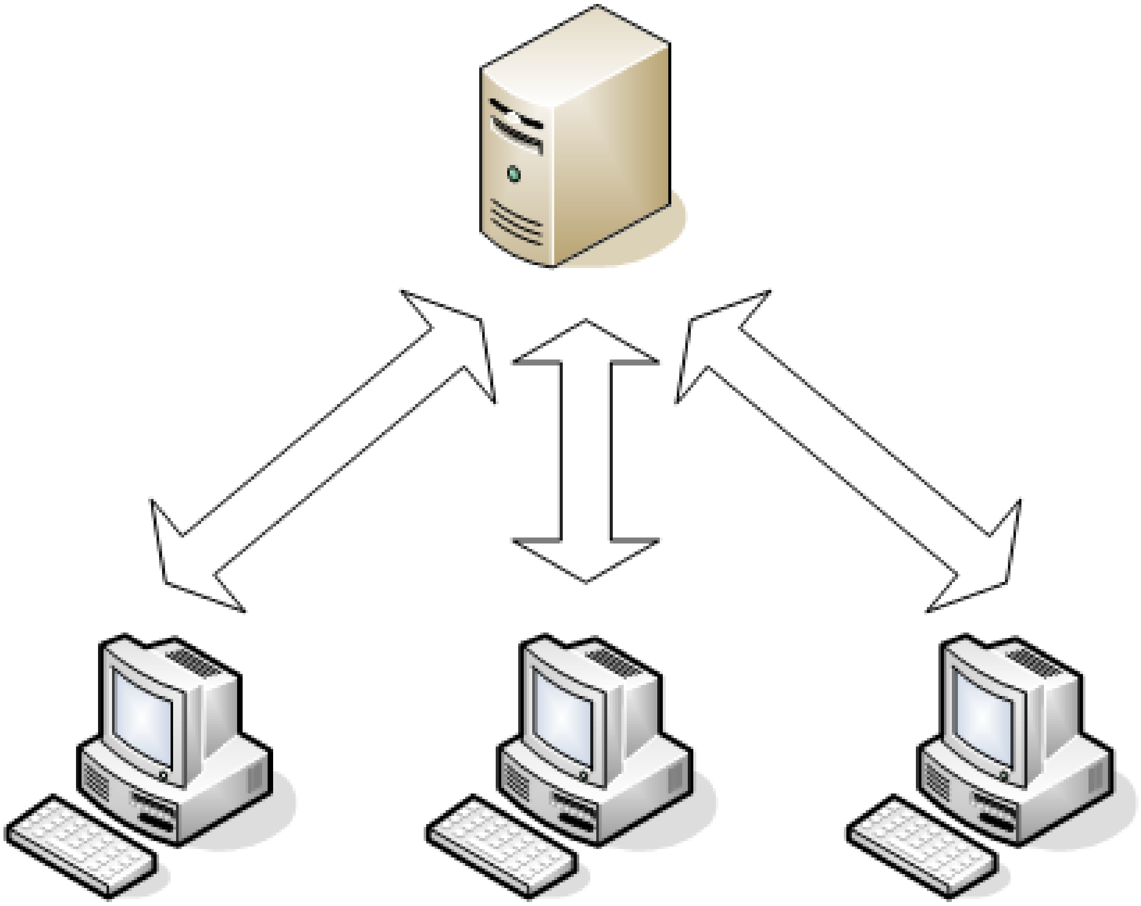


Figure 3.10: Client/Server

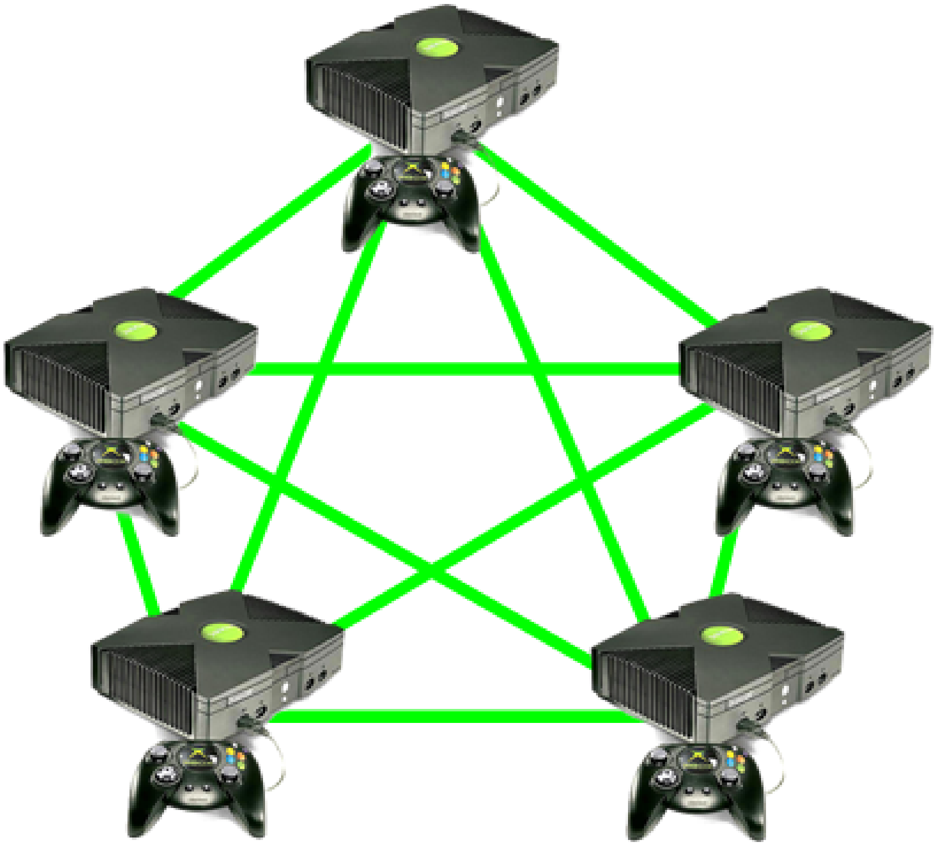


Figure 3.11: Peer to Peer

It should be noted that with this approach that should there be an issue with one of the clients, the effect can be cascaded onto the rest of the clients. Games using the client-server model would need to be using this model even when there are only two active players. One of the machines would act as the server.

Peer to Peer

The second model is a peer to peer connection. (Figure 3.11) This is when clients discover each other and run the game with the information directly from other clients. This model is typically used in the console market. (Consoles being items like Sony's Playstation and Microsoft's X-Box) Many games also use this method for broadcasting and voice communications between players.

3.5 Game Communication Life-Cycle

New section on game lobby plus actual game traffic

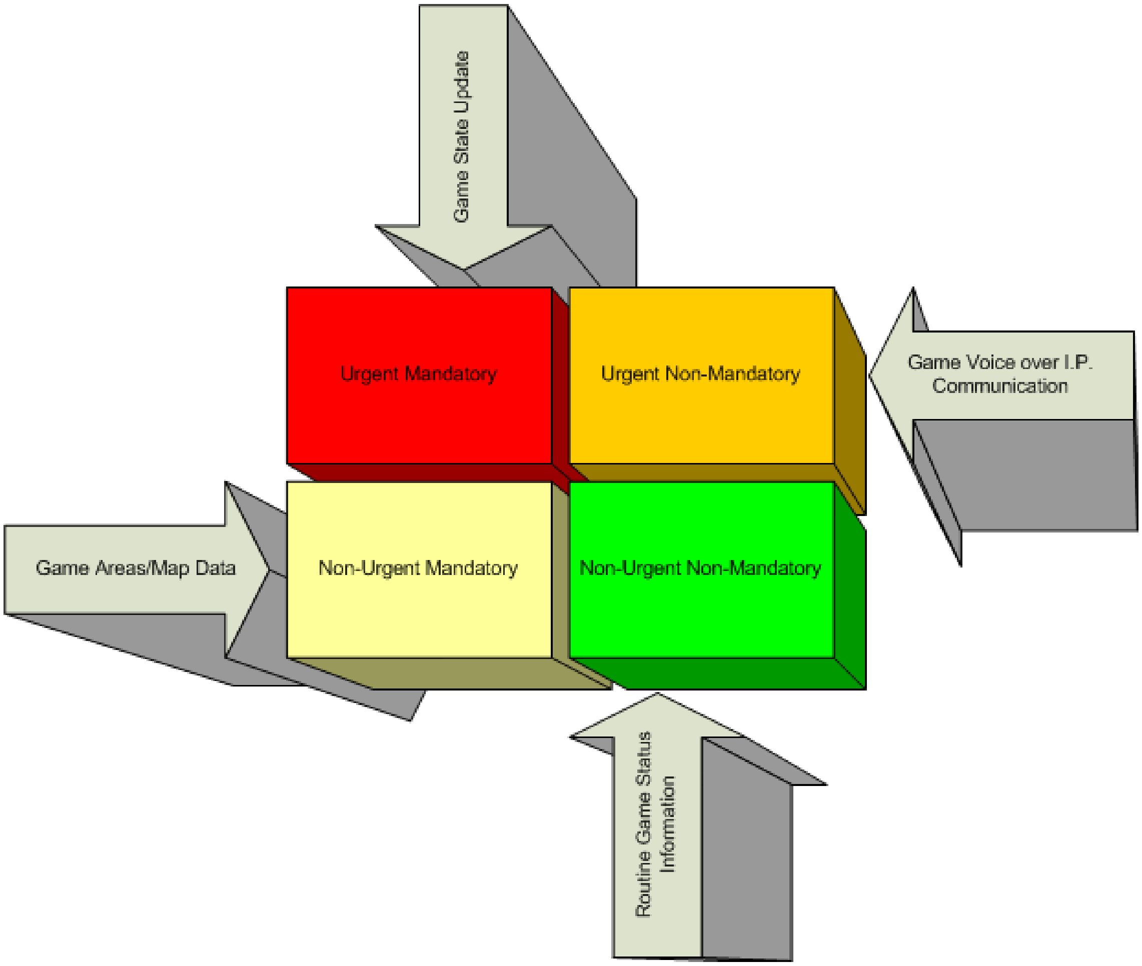


Figure 3.12: Game Communication Types

3.6 Types of Data Communicated

And finally, we come to the types of data communicated by games. There are many types of data communicated, these are used to synchronized the communication between game environments or provide communication between clients.

Game State Update

This refers to the updates given by clients to update the game world represented so that this can be synchronized with other clients. This includes user actions, system events and so on.

Player/Server Communication

This refers to text chatting between users and the server sending system messages. This can be Peer to Peer for between users and client/server for system messages. Some games now support Voice over I.P. to allow for better communication between users.

Administration

This refers to traffic like authentication and billing. This is almost always in the client/server model.

Game Program Updates

As bugs are found and the game balance tweaked, updates to the game software will be required. Typically this is achieved using the client/server model.

Chapter 4

Middleware

4.1 Middleware Description

Middleware can be described as a software engine that provides translation between the backend server software and the front end application. This would be the classical definition of middleware. It allows systems which speak different protocols to understand each other.

An example of this would be ODBC or Open Database Connectivity. ODBC provides a standard programming interface for database client applications to communicate with a number of database server back-ends. These allows for the software developer to write database independent code and the end user to use any data source.

4.2 Gaming Middleware

The term gaming middleware can be very deceptive as it often describes third party or in house software libraries in which game developers can use to form the main bulk of code in a complex game. This is done as many functions and procedures as similar from game to game like graphical routines and input handling that it makes sense to have a base code which would be reused from game to game. This is commonly referred to as the Game Engine.

There are many types of Game Engines available, though the most common type would be that of the graphics engine in which the graphical representation of the gaming simulation is modeled for the screen. Other types include the Artificial intelligence which provides the computer player actions, the physics engine in which the game world is model.

4.3 Specific Gaming Network Middleware

Gaming Network Middleware refers to the sub class of middleware used for multiplayer games. These started out as simple connections in which data is exchange in a simple duplex method. There are many types of middleware to support the different aspects and types of Multiplayer Games. These can be broadly divided to 3 types.

The first type would be that of the communication interface. This type of middleware is used as a low level link between computers. This type of middleware gives the game developer a simplified and standard approach to develop multiplayer games. Features include error recover and in some of them encryption.

The second middleware of note would be the Game Lobby or Matchmaker Middleware. In this type of Middleware, players are brought together in a user friendly method. This middleware allows players to find each other over the Internet or in Local Area Networks.

The last type of middleware would be the integrated game engine. In this type of middleware, middleware developers provide a full programming framework in which game developers can add game content. (Game content would be the game world physics, graphics and game-play) This is mainly use in Massive Multiplayer Games as the programming of the network engine is the most complex.

4.3.1 DirectPlay/xBox Life

Microsoft DirectPlay is a subset of the Microsoft DirectX library of game API. DirectPlay is the most popular middleware engine for multiplayer games. It uses UDP on a retransmit if not received model. It also supports other protocols and game communication channels like IPX or Modem play.

4.3.2 TerraZona

TerraZona is a Windows based Massive Multiplayer Middleware framework in which developers add their own game content to it. The solution is based on Microsoft Windows and Java Technology in the servers. TerraZona hopes to provide a full end to end solution for smaller game developers to deploy their own Massively Multiplayer Products with their managed services.

4.3.3 GameSpy Arcade

Gamespy Arcade is the most popular matchmaking middleware system for Windows Games today. This system allows players to find each other over the Internet. Many games now develop in built support but older games are also supported through their additional client.

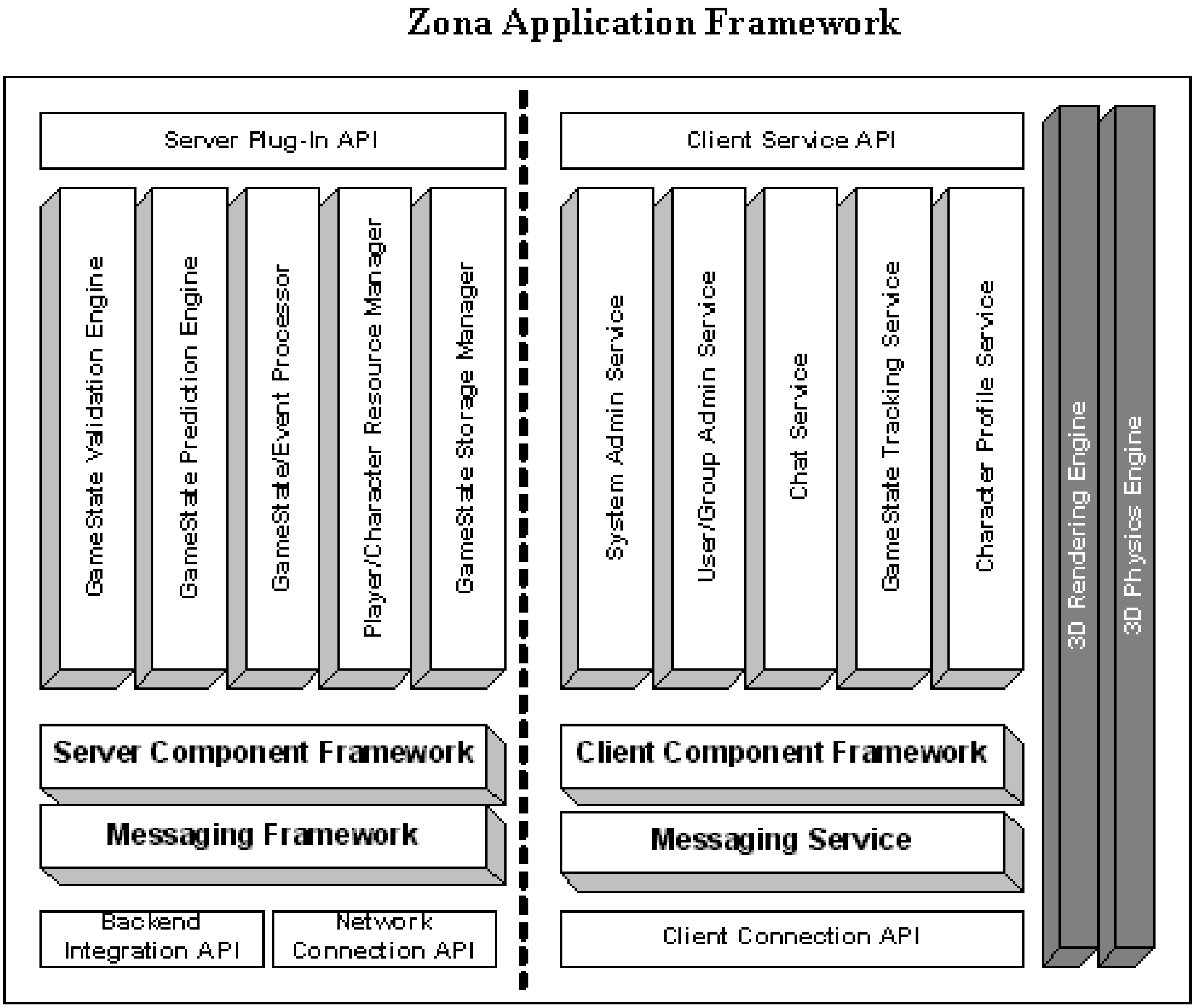
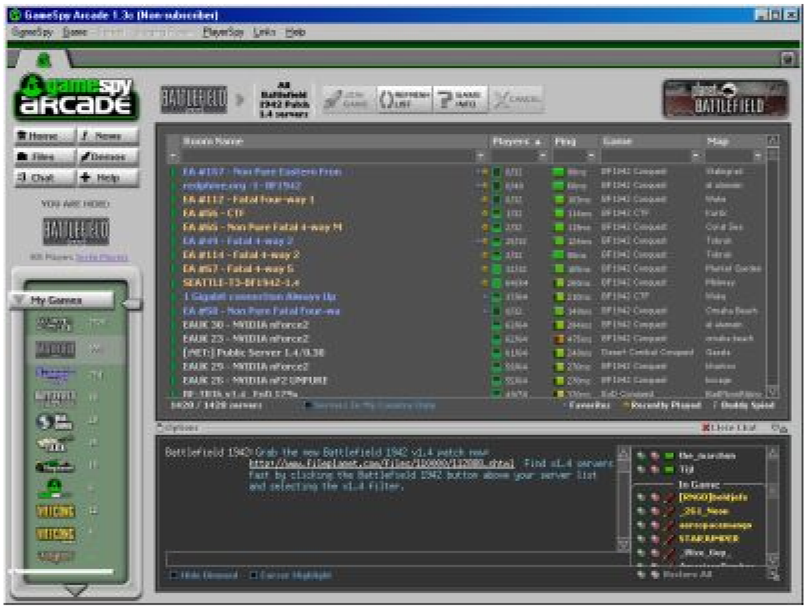

Zona Application Framework
Server Plug-In API
Client Service API
GameState Validation Engine
GameState Prediction Engine
GameState/Event Processor
Player/Character Resource Manager
GameState Storage Manager
System Admin Service
User/Group Admin Service
Chat Service
GameState Tracking Service
Character Profile Service
3D Rendering Engine
3D Physics Engine
Server Component Framework
Messaging Framework
Client Component Framework
Messaging Service
Backend Integration API
Network Connection API
Client Connection API

Figure 4.1: TerraZona Architecture



Room Name	Players	Ping	Game	Map
EA #157 - Non Pure Eastern Fron	[illegible]	[illegible]	BF1942 Conquest	Stalingrad
ecdphere.org :1 - BF1942	[illegible]	[illegible]	BF1942 Conquest	el alamein
EA #117 - Fatal four-way 1	[illegible]	[illegible]	BF1942 Conquest	Wake
EA #56 - CTF	[illegible]	[illegible]	BF1942 CTF	Kursk
EA #66 - Non Pure Fatal 4-way M	[illegible]	[illegible]	BF1942 Conquest	Coral Sea
EA #49 - Fatal 4-way 2	[illegible]	[illegible]	BF1942 Conquest	Tobruk
EA #114 - Fatal 4-way 2	[illegible]	[illegible]	BF1942 Conquest	Tobruk
EA #57 - Fatal 4-way 5	[illegible]	[illegible]	BF1942 Conquest	Market Garden
SEATTLE-T3-BF1942-1.4	[illegible]	[illegible]	BF1942 Conquest	Midway
1 Gigabit connection Always Up	[illegible]	[illegible]	BF1942 CTF	Wake
EA #58 - Non Pure Fatal four-wa	[illegible]	[illegible]	BF1942 Conquest	Omaha Beach
EAUK 30 - NVIDIA nForce2	[illegible]	[illegible]	BF1942 Conquest	el alamein
EAUK 23 - NVIDIA nForce2	[illegible]	[illegible]	BF1942 Conquest	omaha beach
[MET:] Public Server 1.4/0.38	[illegible]	[illegible]	Desert Combat Conquest	Gazala
EAUK 29 - NVIDIA nForce2	[illegible]	[illegible]	BF1942 Conquest	[illegible]
EAUK 26 - NVIDIA nF2 UNPURE	[illegible]	[illegible]	BF1942 Conquest	bocage
BF-1918 v1.4 ...	[illegible]	[illegible]	[illegible]	[illegible]

1420 / 1420 servers

Battlefield 1942: Grab the new Battlefield 1942 v1.4 patch now: http://www.fileplanet.com/files/110000/112000.shtml Find v1.4 servers fast by clicking the Battlefield 1942 button above your server list and selecting the v1.4 filter.

Figure 4.2: Gamespy Arcade Screenshoot

4.3.4 Demonware

Demonware is a network communication middleware which supports cross platform between the PC and Game Consoles. It uses an encrypted UDP channel to communicate between clients. It also has its own matchmaking middleware software.

4.3.5 Quazal Net-Z

Quazal Net-Z is a network communication middleware system for between 2 to 32 players using an object passing model instead message passing. Features include Play Station Portable compatibility and it has it own match-making middleware called Rendez-Vous. Game Developers include Red Storm Entertainment makers of the famous Splinter Cell series of games.

4.3.6 Massiv

The Massiv is a distributed game middleware whose purpose is to simplify the development of massively multiplayer online games. As an Open Source project, it has many appeals. Currently, there are no takers in using this middleware but it provides a graphical demo in order to show the use of this software.

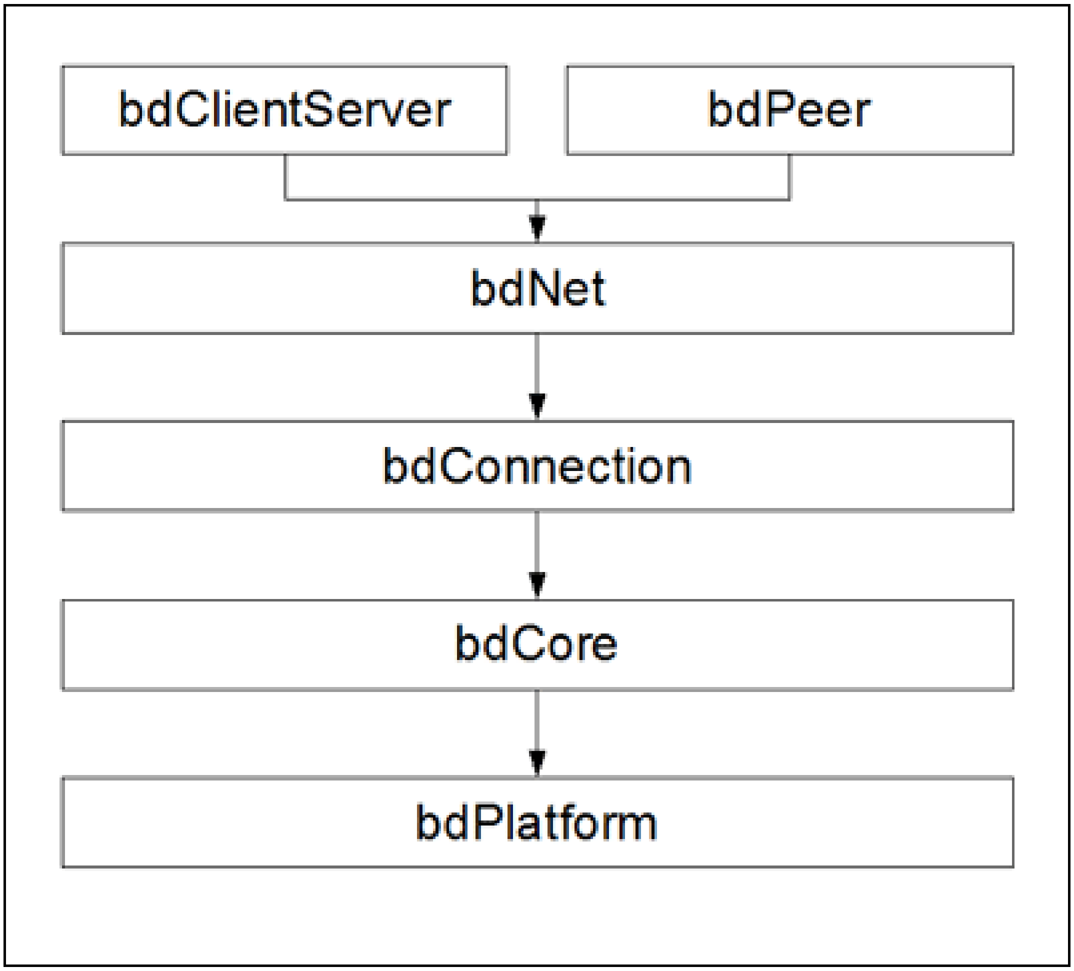


Figure 4.3: Demonware Middleware Architecture

4.3.7 MUPE

MUPE or Multi-User Publishing Environment is developed by Nokia as a Game Communication Middleware for Mobile phones. This middleware uses the HTTP Protocol as the primary method to transfer data between client and server in a push and pull system. A main disadvantage of this would be the speed of the system. However, this supports the MIDLET 1.0 Profile which is the lowest common operating platform for all Java Mobile Phones.

Chapter 5

Requirements for Game Communications

In the search for the best solution for the game communication problem facing designers, we must decide on what are the requirements of game software and game developers. In this chapter, we identify some key communications requirements to support a multiplayer game. Some of these considerations might not seem important or even trivial, but in order to support an engaging game, most if not all must be met in the design process.

The concept of fairness and balance also comes into game play. This is because in order to have a fair game in which no one player has an advantage over the others due to network or hardware differences. Players do not like items like phantom bullets where due to a game state update being delayed, a “killed” character is able to act due to the fact that that client is not informed of the new state until after that action.

5.1 Reliability

The first item we need to consider is reliability, and in particular the ability to deal with failure in the network.

Recovery from failure is an important issue in the implementation of a gaming communication protocol. This is because almost all channels of communication between the client and server will have multiple potential points of failure.

Networks may have different levels of reliability at different times. One of the design features of TCP is that it is capable of slowing down or speeding up depending on the state of the network. Therefore, simply by using TCP/IP we are able to provide a more reliable game communication service.

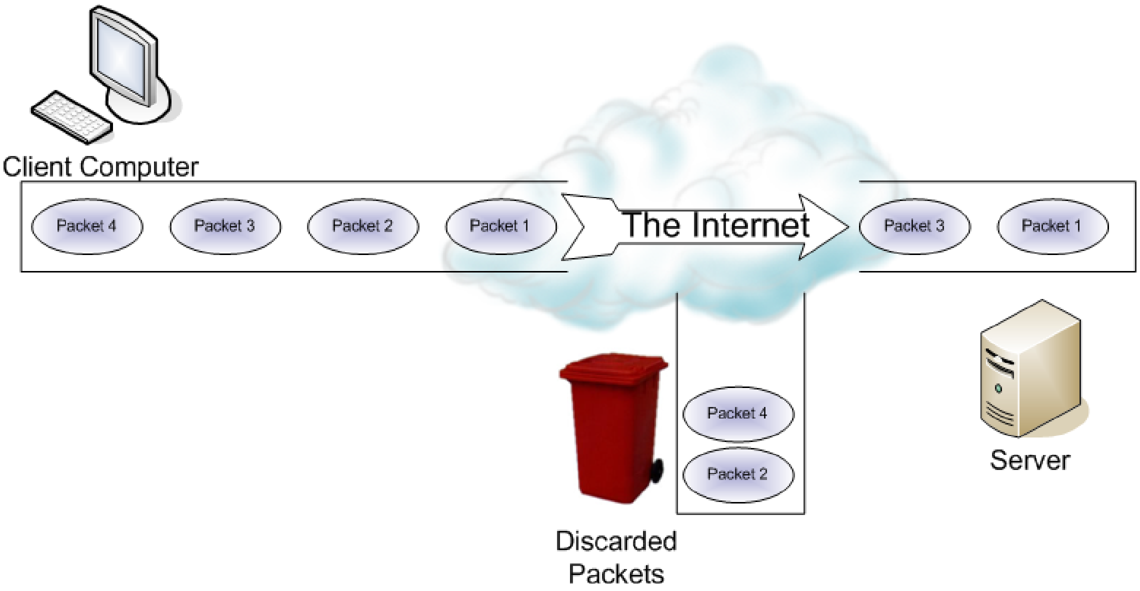


Figure 5.1: Packet Loss

5.2 Quality of Service

Throughput

Throughput is the amount of data that a communication channel can carry. The appetite of a game for throughput should be kept low in order that it can continue to function under the widest possible range of conditions. The opposite is in effect for the clients in which they want to get as big of a pipe as possible.

This is important as in order give a high level of throughput for games, business have setup Internet Cafe type operations where users rent computers just to play games. The term describe for this is “Lan Shop”. In such an establishment, reasonably powerful computers couple with a low latency network (Ethernet 100 Base-T) is used to provide for the type of throughput needed for these types of games.

Another interesting point would be that of “LAN Parties”, these are events in which participants bring their own computer to a location to play such games.

On the other extreme of the spectrum, we notice the cost of communications via Mobile Networks to be still rather high for data communications. Thus it makes a lot of sense to lower the throughput requirements to that which is acceptable to the budget of players.

The key here is efficiency. A protocol design for this must be efficient enough to support itself but remains functional for the game it supports.

Loss

Packet loss is when a data packet disappears and does not arrive at its destination.

It is important to measure the rate of loss in order to establish the quality of service of the network. A high amount of loss indicates serious problems with the network while a low level is acceptable. It should be said that loss cannot be eliminated completely and a low level of loss is normal in the Internet. The two ways to deal with lost packets are: *to retransmit until it is received*, or *ignore the packet completely*.

Delay

There are four main types of delay, propagation delay, transmission delay, queueing delay and processing delay. We have limited control over most of these however we do have some ability to control delay due to queueing. Queueing delay at buffers on the path to the server, and also at the games server are both potentially a problem which could be alleviated by reducing the volume of packets sent. Hence, if we can find a way to send more data when conditions are good and less when conditions are bad, it could be an advantage in the control of delay.

Delay is commonly refered to in gaming circles as lag and is typically measured by the ping rate between client and server. This actually measures round-trip-time, i.e. the delay between a packet being sent and a reply being received. Note that this includes processing delays at the server. The main point to make here is not to add to the problem of delay. This can be done via reducing the need to transmit and to do it only when needed.

5.3 Processor

A Computer Game can be a very complex application which takes up lots of resources. There may be many types of resources, but in this section we refer mainly to that of the CPU.

There are two types of computers in discussion here, the client and the server. All computers have limited processing power. The company hosted and owned server has more strategies involved in scaling upwards to meet the demands of the player base by spending more capital. This is done either by making the code more efficient or buying more servers. This is referred to by some members of the industry as a “happy problem” as it would mean that the business of the actual game is booming, thus creating the need for more processing resources. However, the player hosted server would fall under the same requirements as that of a client.

Traditionally, the biggest load on the processor would be that of the graphics engine. Still a major component, most of this work on the PC and Console platforms are now off loaded to the GPU. (Graphics Processing Unit) This dedicated processor has a specific instruction set for processing graphics especially 3D graphical calculations.

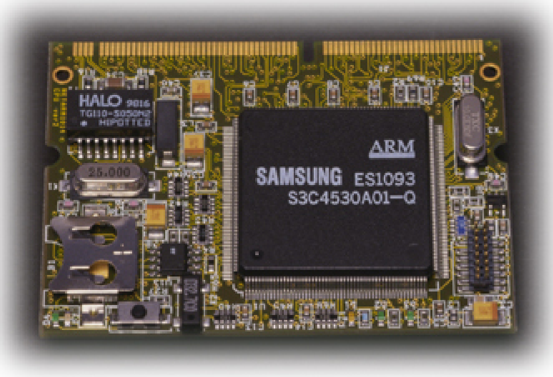


Figure 5.2: An ARM Processor

The next item of note contributing to processor load would be that of Artificial Intelligence. This refers to the ability to in an “intelligent” way game’s software reaction to the player’s actions. This is fast becoming the main processor hungry point in the game’s resource requirements.

With the 2 above needs, we still have to squeeze in items like the game world simulation engine (Physics and Environment), audio capabilities, operating systems, user input and other requirements. It is no wonder that communications is not taken into consideration in a major way.

With all of this processor load on top of the current and next generation desktop and console systems, one would wonder if the crop of mobile devices can keep up. The typical mobile device now use a Advanced RISC Microprocessor with a internal clock speed that is a fraction of the above mention devices. We can observe that the games of yesteryear had to deal with the same limitations but the consumer demands the same standard of entertainment pleasure. (The Palm Pilot System only used a 33Mhz Motorola Dragonball VZ processor for its top of the line m515 PDA before switching to a ARM Processor for later systems.)

The inference from this section is that the communications API and Protocol must not take a significant amount of processing power. In fact, it should be kept to the absolute minimum in order to preserve resources for use elsewhere.

5.4 Responsibility

Designing a responsible communication systems can be divided into 2 parts, ethical responsibility and network responsibility. We need have to leave it to the end users and network administrators to decide on the ethics of the where and when to use game software. (For instance, playing computer network games in the office environment.) But we as developers can influence the network responsibility.

Network Responsibility means to be a “Good Network Citizen”. This is done in a few ways, chief among which is not to add to the current problem whatever it is. Responsible protocol design dictates that you do not retransmit unless you have a congestion avoidance strategy.

The use of UDP therefore means that you would not retransmit. Game Developers would want to use UDP in their systems due to their high throughput performance but sometimes poorly implement it [GZBS02]. Despite the need for retransmission, they still use UDP. The use of TCP is better because it has a congestion avoidance strategy for such data packets which must arrive at their destination.

5.5 Network Compatibility

The term “Network Compatibility” is highlighted here for a number of reasons. It may seem trivial to say that we should support the most common protocol in use but there are many layers of Compatibility.

The first issue is that of Network Address Translation. This can be termed as a firewall or routing. NAT refers to the usage of a single public I.P. Address shared among a number of computers using private addresses. A packet would be sent from one of these internal private computers with its I.P. address to the router or gateway. This gateway would then replace the private address with its public one and make a note on a table for referencing. Once the reply comes in, the incoming packet is checked against the table and forwarded to the correct computer by the router.

The use of NAT is very widespread, especially in broadband networks. Names like LinkSys and D-Link Broadband routers dominate this market to “share” a broadband connection with more then one computer. Leading Industry Sources recommand this as a method of protection, even on a single computer installation. This also comes into the wireless field where many providers and companies use NAT in this application.

Most applications do not support NAT well. Poorly setup streaming servers cannot piece through the NAT “firewall”. While some applications would support NAT, others fail to consider this popular technology. (Chief suspects are V.o.I.P. solutions.)

The other issue of note to bring up is connection failure. Many links fail and this means that a new connection must be establish. When a new link is created, a new I.P. Address will probably be used. This with NAT in combination would mean that the route to the server will be lost and thus failing the game.

These two issues must be considered before any large scale game implementation over the internet is created.

5.6 Platform Compatibility

Each platform represents a number of design comprises. From battery life versing processing power on the mobile platform to that of the limitations of the development enviroment.

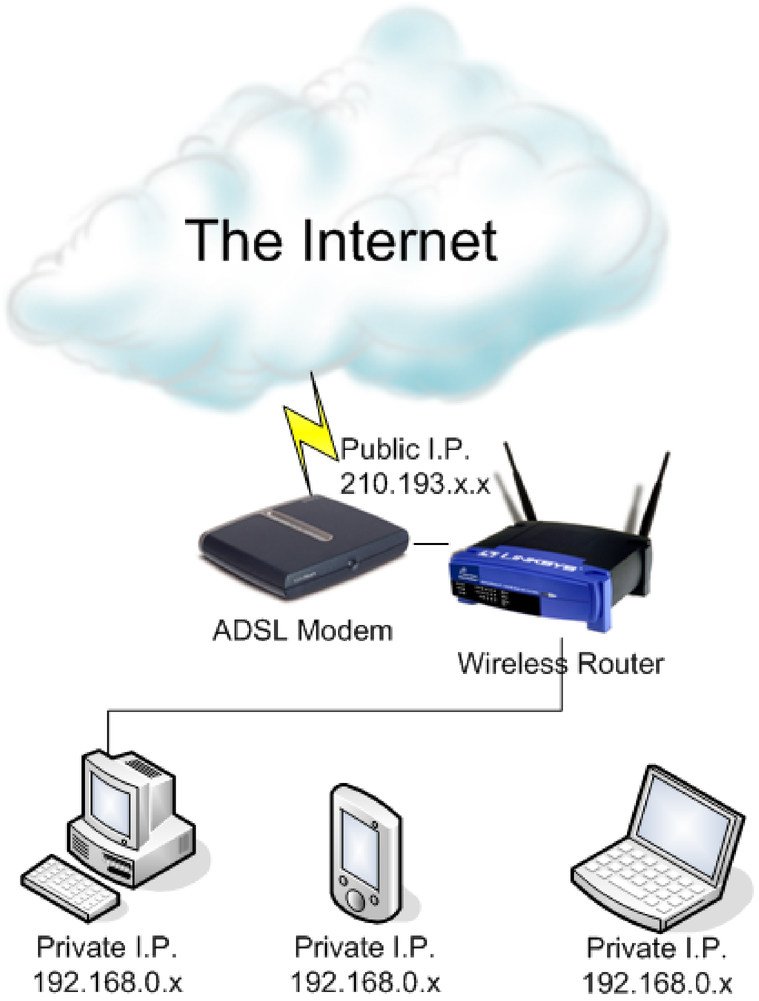


Figure 5.3: Network Address Translation

The Palm OS for the PDA and smartphone in their API can support the full range of socket API but is limited to 4 sockets [Pal04]. This is to reduce the memory requirements.

Another Platform Limitation to be aware of is Sun's J2ME (Java 2 Micro Edition) limitations. The Midlet 1.0 interface only supports HTTP as the communications protocol [Mic00]. The newer standard, Midlet 2.0 implements UDP. [Mic02] This is a concern as not all mobile phones which have java supports the later standard.

Another consideration is that most mobile platforms don't have that much memory or internal/external storage and in relationship to the processor, most mobile platforms don't have a GPU. (Notable exceptions include the Nokia N-Gauge & Tapwave Zodic) These design considerations have a major impact not only on the protocol design but of the game itself.

5.7 Security

Security is an important area in the field of design. Key among them is keeping the game up and fair. Keeping the game up can be left to the security methods that apply to keep servers up. However the issue of keeping games fair is a major concern. [Way03]

Developers should keep in mind that are there people who are willing to exploit bugs or low security in a game to gain an unfair advantage over others. This can be done through the sending of false packets or scripting. Attacks of this nature doesn't go well with the company's image or other players out there to have fun.

Data from Client Computers should always be treated as suspect data until verified. An encryption pipe between the client and the server maybe required in some instances. The obvious way to do this is to connect via a Virtual Private Network (VPN). But the full range of functions of a VPN may not be required. Although total security cannot be successfully achieved in all cases, designers should always endeavor to secure their systems.

Chapter 6

Proposed Gaming Protocol

Once the requirements are identified, the problem should be analyzed. As we can see in the previous chapter, there are many constraints that have been identified for as applying to the protocol implementation. Understanding the impact of these constraints we should be able to find the right solution.

6.1 Proposed Solution

The proposed solution is a *dynamic*, *stratified* protocol. The protocol dynamically adapts to the changing bandwidth capacity of the network. The term *stratified* means that the payload in the protocol packets is subdivided into different categories, or strata, of data. In our case, the overall division is shown in Figure 6.1. This is possible because of the different types of data communicated within the games. By stratifying or having different layers of data, it is possible to control the amount of data being sent out.

6.1.1 Game Data Grouping

As described earlier, game communications data can be divided into different types. We can classify data according to its performance requirements in two ways. The way of classifying data is by its urgency, and by whether the data *must* be received. We shall make use of the term *mandatory* for data which *must* be received for correct game operation. Other data will be referred to as *non-mandatory*. Non-mandatory data may be discarded.

The second method would be that of the urgency. This could be divided cleanly using levels. Data with a higher level must be sent and received very quickly while those at lower levels could be delayed. If we reduce the number of levels to two, data can then be referred to as urgent and non-urgent data.

From the two dividing methods we can draw Figure 6.1. All types of game communication can be reflected by the matrix. Key Game Updates would fall under the

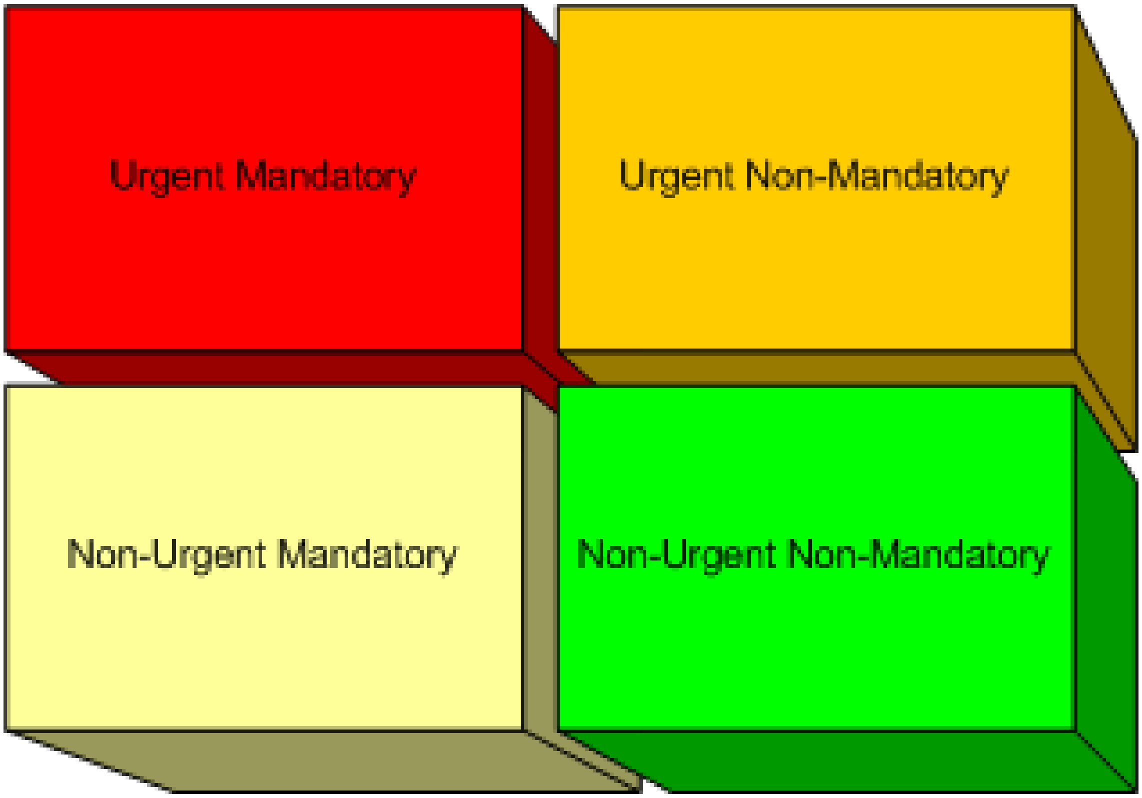


Figure 6.1: Game Communication Matrix

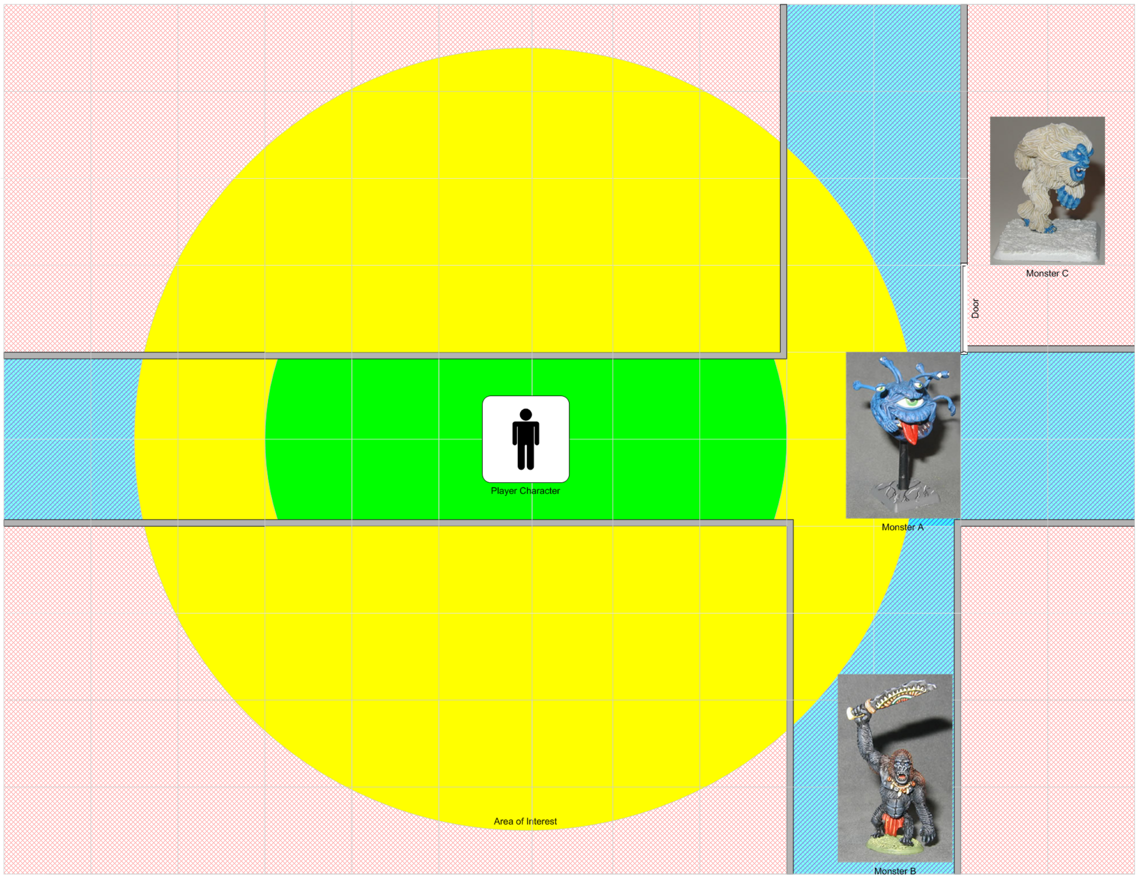


Figure 6.2: Area of Interest for Stratification

Urgent Mandatory Sphere while Voice over I.P. communication would fall under the Urgent Non-Mandatory Sphere. Game Patching and Updates would fall under the Non-Urgent Mandatory sphere.

Game State Update Stratification

While the other types of data are important. The most bandwidth consuming part of any game would be that of the Game State Update. This is a key area for Stratification.

With reference to Figure 6.2, we can see how this could be stratified. Firstly the Figure would refer to the environment in which a player of the game is in. The character that the player controls is that of the "Player Character" represented by the White Box and Man Shape. The Green Part would represent the field of vision that the Player has in this environment. The blue portion would refer to that of movable area in which the Player Character can move. There are 3 monsters in this environment.

The key here is the yellow circle centered on the Player Character. This would represent the "Area of Interest" in which Game Updates should be sent to the client. By defining an "Area of Interest" the game server doesn't need to relay all the information of the current map to the client. As we can see a Single Monster is in this Area of Interest. Information about this monster is transmitted over to the client. While the other 2 monsters are not in this Area of Interest and therefore do not be required to be transfered to the client.

The size of this circle can depend on a number of factors. The best approach would be that of a dynamic circle size with the smallest to be that of slightly larger then the player’s field of vision. This size would then grow bigger when there is enough bandwidth to support the increase in size. The decision to increase the circle size is dependent on the performance of the network. This is the dynamic adaptive part of the protocol.

6.2 How does a Stratified Protocol Address the Issues

By assigning different levels of service. The communication system can decide on what type of data to be sent instead of sending all the data as and when needed. It can delay certain packets until the network can support it.

By using a Stratified Protocol, developers can address the issue of Quality of Service and Reliability. This approach is highly responsible in which data is transmitted only when required and the network can support it.

6.3 Problems Not Solved using the Stratified Approach

Security is one issue not solved by the Stratified Approach. This can be left to the developer to implement as part of the system by encapsulating the data with encryption before stratification. [Ise02]

The other issue not solve using stratification would be that of bandwidth. Stratification cannot increase the bandwidth capacity of a channel but it can make more efficient use of the bandwidth provided.

6.4 Roaming and Recovery

Wireless Channels of communication are known to be prone to failure for what ever reason. When there is a failure it is important to get back as soon as possible. But when getting reconnected, you might not be having the same I.P. Address as before. This is even more true when you are on a traveling machine. The connection point can change as you roam from one place to another.

This issue, we can refer to as *I.P. Address Hopping*. It is important on the server end (Since this is a fixed point) to keep the possibility of reconnection during a connection failure. The client can communicate the new I.P. address to the server using a unique client I.D. When the server receives this new I.D., it can resend the mandatory packets to the client using the new I.P. address.

6.5 The Unified Approach

No one single solution can satisfy all the requirements set out. Such a solution may not be the best approach. But by using a combination of techniques shown in the above we would be able to meet the requirements set out in the last chapter.

Chapter 7

Design

A Good Design allows expansion and reliability. Once programs are coded, it is harder to change the functionality of the system. A Stratified Network Protocol Approach was proposed in the last chapter. This chapter describes how the protocol can be implemented.

An Application Program Interface (API) provides standards for the game developer. By using standards, it separates the communication library implementation from the game which can be developed separately. By using a separate library provides for the code reuse in other software development. Improvements can flow back to the original software when the API implementation is updated.

7.1 The TCP and UDP API

We need to use both TCP and UDP in the design. This is because each has its strengths and weaknesses. In Figure 6.1, there are two main classification on the horizontal axis, namely is Mandatory and Non-Mandatory.

TCP should be used for all Mandatory Data because UDP must not be used for data which will be resent. This is an inviolable principle of responsible network protocol design. UDP on the other hand should be used for Non-Mandatory Packets as the network can discard them when the need arises.

As for the Vertical differentiation, it is possible to use the Type Of Service field which in the I.P header. to give the protocol the different levels of stratification. There is no consistent use of this across the Internet. This might become an option in the future. However, already the Type Of Service is used to provide performance differation in internal networks and routers.

In order to simplify the task of application developers, a common API should be developed to handle communication for all strata. This would act as a “datagram” type socket but would be implemented using both TCP & UDP. A Datagram message is used as to allow easy differentiation of types of data and in game programming communication normally occurs in messages. In Addition to this, most developers in

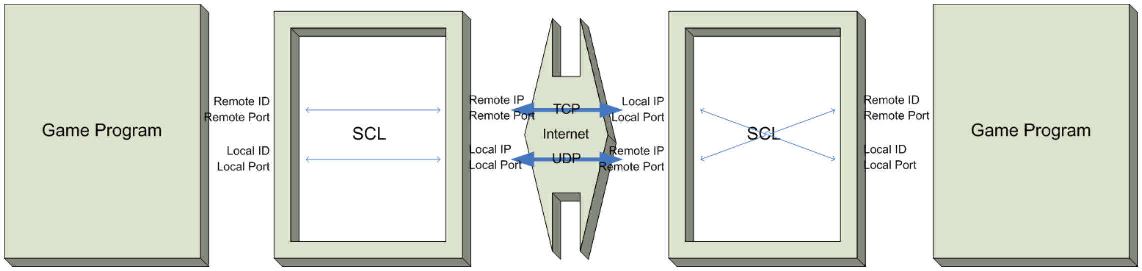


Figure 7.1: API Connection

their haste to provide faster communication currently use UDP with a retransmission function for lost packets, so they clearly know how to use the datagram interface. [GZBS02]

As we can see in Figure 7.1, the use of TCP and UDP is encapsulated in the *Stratified Communications Library* (SCL) Black box. It makes use of 2 links to the opposite side via both the TCP Channel and a UDP Channel then through the SCL API provides a single unified datagram interface. As the I.P. Address is also encapsulated in this design, it is possible to implement the I.P. hopping concept presented in earlier chapters.

7.2 Buffering

In order to provide for better use of bandwidth, when the state of the network is busy, packets of a lower stratification in priority should be buffered. The congestion status of the network could be measured by monitoring the value of the TCP congestion window and the round trip time delay. This would indicate the available bandwidth of the network in question.

When the heath is good, packets would be sent accordingly but when the performance of the network degrades, packets should be placed in a queue and served in priority order. Top Strata Packets would not be queued at all but sent as soon as received.

A seperate thread should be used to service the buffered packets. Packets would be serviced in strata priority order. This process would depend on the number of urgent packets which needs to be sent, and it would send out more packets when the performance improves. This would be in the order of Stratification and first in first out. That is packets of the higher strata get sent first and within a stratum packets would be sent First In First Out.

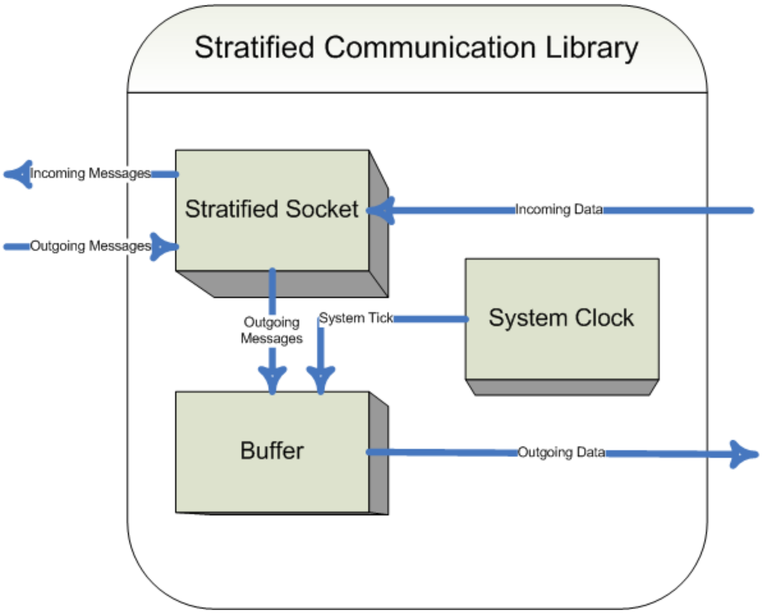


Figure 7.2: SCL Block Diagram

7.3 Packet Discard

Packet Discard only applies to non-mandatory packets, with UDP using the TCP congestion window to gauge the network status. Certain non-mandatory packets may be queued. However, at some point in the lower levels of stratification Packets would be discarded by the sender without even transmitting them over the Internet.

This would be done from the lowest strata upwards depending on the network conditions. With the above process trying to send out packets, this would be implemented as another thread. This would remove the non-mandatory packets from the buffer after a certain period of time. This would be done on a first-in, first-out discard policy in which the oldest non-mandatory packets in the buffer of the lowest strata gets discarded.

Game State Update Messages can be grouped into a series describing a certain discrete event. In such a scenario, the best approach to make efficient use of bandwidth is to use different levels of stratification for the different messages. (Figure 7.4) The trigger event would be sent using the highest mandatory stratum while updates be sent using the lower levels in the non-mandatory stratum. The final update should be then be sent using the top mandatory stratum. This method allows the communication library to handle which packets gets transmitted and at what point.

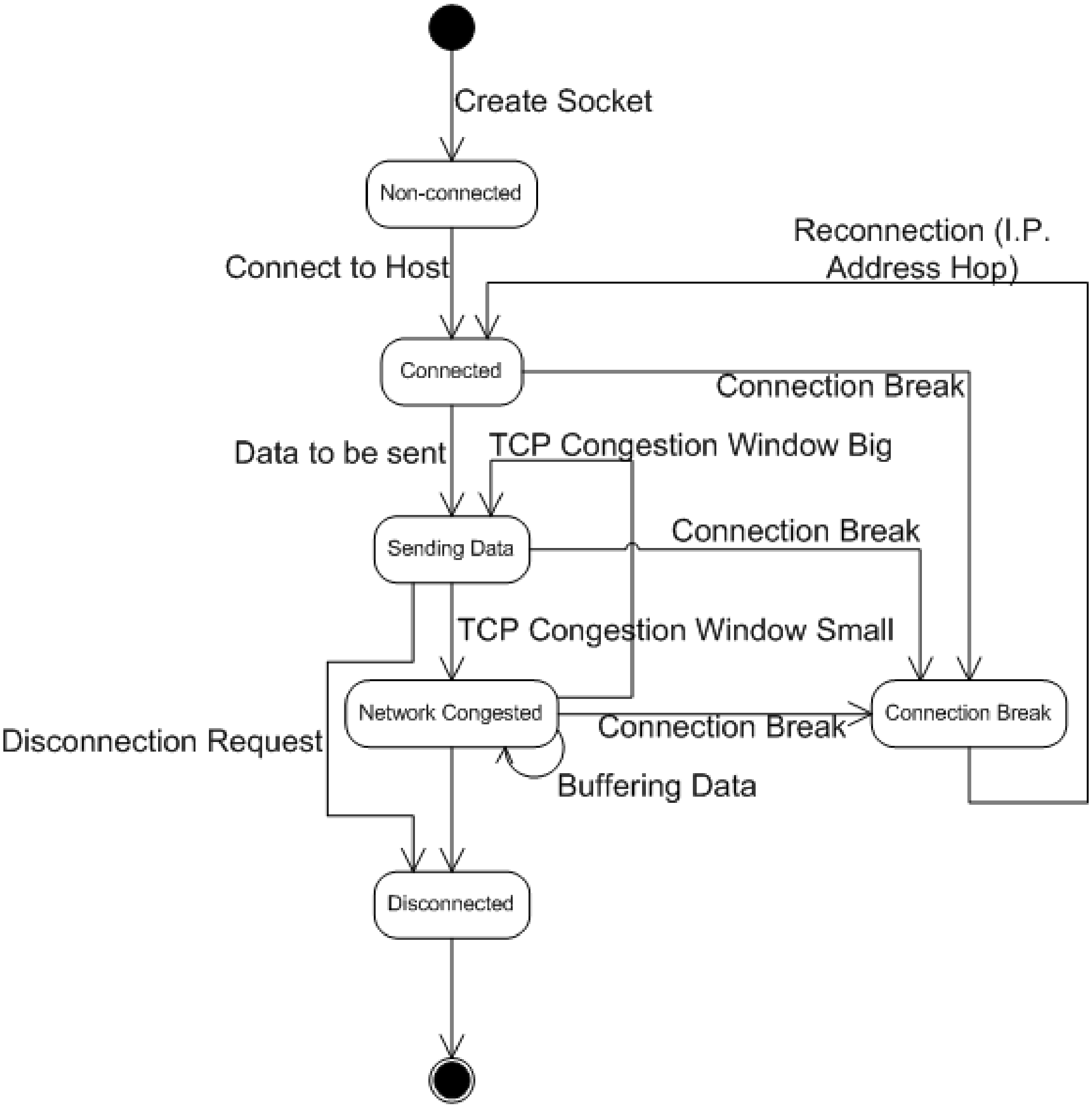


Figure 7.3: SCL State Transition

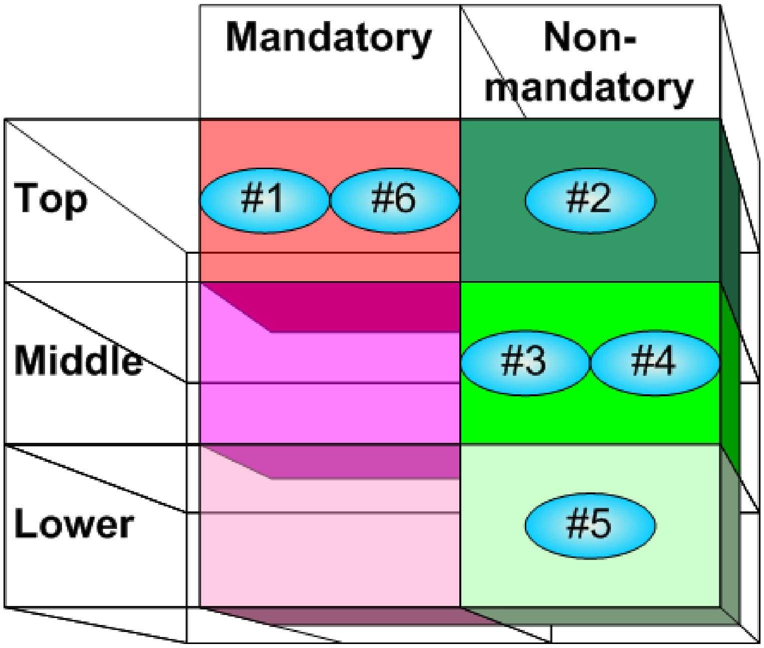


Figure 7.4: Stratification for Message Series

7.4 I.P. Address Hopping

On the server end, it is important to point the packet to the correct process handling the client should the connection be disconnected. An Active table of clients would be kept by the accepting process and when the client id gets received this would be checked against this list.

Should the client be that of an active user, the connection would be forwarded to that process which took care of that client. It should be noted that an idle active process should be able to expire after a certain amount of time to clear the use of resources. Any game related cleanup operation would be done at this stage. This is of course to provide for a smoother transfer of I.P. Addresses.

Chapter 8

Player's Satisfaction under Network Stress

To investigate the relationship between network performance & game playing satisfaction, an experiment was performed. In this experiment participants played a computer game under changing computer/computer network performance conditions. During the study, information was gathered concerning the network and after each session the experiment players were surveyed to gather information on their gaming satisfaction perception. By doing this experiment, we can establish the parameters in which a stratified protocol can be measured against a non-stratified protocol.

The purpose of this experiment was to investigate the following hypotheses.

8.1 Hypotheses Considered

Hypothesis 1

Network Conditions have a direct impact on Game Player satisfaction

The player satisfaction quality perceived by participants changes when the network conditions change. In this chapter we shall assume that player satisfaction is a function of the network conditions. We shall refer to this relationship as the response curve. Refer to §8.9 for more details. Hypothesis 1 was confirmed. See evidence in §8.9.1.

Hypothesis 2

Response curves are different for different Games.

The response curve is different for different games. The response curve is characterized by a collection of coefficients. Therefore, we concluded that the response curves are different by observing where the differences between the coefficients are statistically significant. Examples of estimated response curves are shown in Figure 8.1. The

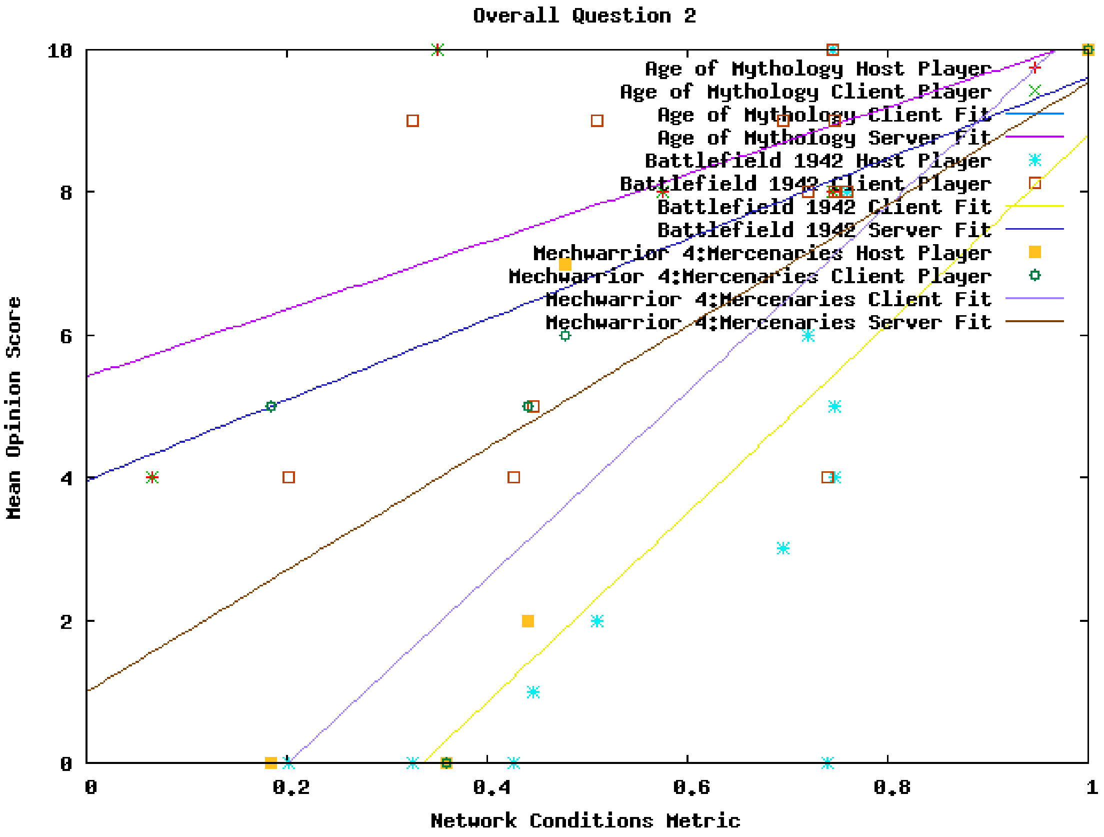


Figure 8.1: Response Curve Example

response curve is actually a straight line in all cases considered here, however in a larger experiment, where more complex models could be fitted, this might no longer be the case. Hypothesis 2 was confirmed. See evidence in §8.9.1.

Hypothesis 3

Different Genres have different Network Requirements

The network traffic patterns are different between the different games. Refer to the Appendix for the traffic plots. Hypothesis 3 was confirmed. See evidence in §8.9.1.

Hypothesis 4

Because Game Traffic varies between games and network conditions, protocol stratification is useful.

This cannot be proved at this stage, however the evidence is supportive of this hypothesis. Hypothesis 4 was not confirmed. See details in §8.9.1.

8.2 Supporting Literature

This experiment is modeled after the paper “Subjective quality assessment for multiplayer real-time games” [SERZ02] but we included a few games instead of selecting a single game as described in that paper to compare and contrast network performance under stress. It should be noted that there are studies that document the results of poor network performance and its effect on the player. [BCL^{+}04, QML^{+}04, SGB^{+}03] But the new ground here is to try to apply the same subjective assessment via Mean Opinion Scores to a variety of games and compare the effect of poor network conditions on player’s response.

8.3 Model

The model of user behavior is that perceived quality is random but has a mean value which is a function of technical conditions. These technical conditions are expressed as a metric made up of 4 variables in equal weighing consisting of Delay, Bandwidth, Loss & Duplication. Due to the small size of the experiment, no attempt was made to model response to individual parameters describing delay, bandwidth, loss or duplication. The perceived quality is ranked in the form of a Mean Opinion Score. The four components of the aggregated statistic are the user settings in the software we used, Nist.Net. Delay in this model is the delay added to each packet as expressed in milliseconds. Bandwidth is express as the maximum number of bytes which can be transmitted in a certain interval. Loss is the percentage lost of packets transmitted and Duplication is the percentage of the packets duplicated. These values are then normalized so that they would fall in the same range and then averaged. The formulas as described below.

The Delay Metric is calculated as below

$$1 - \frac{\text{Delay}}{500} \tag{8.1}$$

The Bandwidth Metric is calculated as below

$$\frac{\log \frac{\text{Bandwidth}}{\text{Highest Bandwidth Actually used in all Games}} + \text{Lowest Result converted to Positive}}{\text{Lowest Result converted to Positive}} \tag{8.2}$$

We use the value of 1 if there is no bandwidth limitation set.

The Loss Metric is calculated as below

$$1 - \left(\frac{\text{Loss}}{100} - 0.9\right) \times 10 \tag{8.3}$$

The Duplication Metric is calculated as below

$$1-(\frac{\text{Duplicated}}{100}-0.95)\times 20 \tag{8.4}$$

The end result of equations (8.1) to (8.4) is to produce four metrics, one each for delay, bandwidth, loss and duplication, each varying over the range 0 to 1, with worst quality for the lower value of the metric. The overall Metric which was used in the analysis of results was the following:

$$\frac{\text{Delay Metric}+\text{Bandidth Metric}+\text{Loss Metric}+\text{Duplication Metric}}{4} \tag{8.5}$$

Which is a simple average of the four metrics. This overall measure of technical quality also ranges from 0 to 1 with lower values representing worst quality.

8.4 Games Selected

In order to show a difference between the requirements of different Genres, we selected 3 Games from 3 Genres, being First Person Shooter, Action Simulation and Strategy. Listed under the games we also added the recommended system requirements to gauge the game.

8.4.1 Battlefield 1942

Battlefield 1942 is an expansive first-person shooter (FPS) set in World War II developed by Digital Illusions CE and published by Electronic Arts for the PC (2002) and Macintosh (2004). The game can be played single-player against bots (or cooperatively with other humans and bots versus other bots), but most of the focus has been on its support for large-scale, multiplayer Internet games. On average, there are 1700 or more servers running Battlefield 1942, with up to 64 players playing on each.

System Requirements

- 800 MHz CPU
- 256 Megabytes of RAM
- 3D accelerated 64 MB video card or equivalent with HTL and a 24-bit z-buffer
- 160 MB free hard disk space plus space for saved games
- 16X Speed CD-ROM/DVD-ROM
- DirectX 8 Compatible Sound Card

- MS compatible mouse
- Keyboard

Multiplayer Features

- Dedicated Server
- DirectPlay
- UDP & TCP Traffic

8.4.2 Mechwarrior:4 Mercenaries

Mechwarrior:4 Mercenaries released in 2002 is a game in which players pilot 100 ton Mechs (Walking Tanks) in a future setting of 3063. Developed by Cyberlore Studios for the Microsoft Games Studio.

System Requirements

- Windows 98/ME/2000/XP
- 128MB Ram
- 700Mhz
- 1 GB Hard drive space
- 16MB Video Card
- 24x CD drive
- Sound card & speakers/headphones

Multiplayer requirements

- Multiplayer 56kbps modem or LAN
- Maximum 8 players with narrowband
- 16 players with broadband
- Modem speed 56Kbps
- Internet Explorer 5.5 and above

Multiplayer Features

- Dedicated Server
- DirectPlay

8.4.3 Age of Mythology

Age of Mythology (sometimes abbreviated AoM) is a real-time strategy computer game in the Age of Empires series by Ensemble Studios. It was first published in November 2002 by Microsoft Game Studios. Unlike its predecessors, Age of Empires and Age of Kings, Age of Mythology has less of a focus on historical accuracy. Instead, the game centers on the myths and legends of the Ancient Greeks, the Ancient Egyptians, and the Norse, allowing players to not only control the historical aspects of these three great civilizations (such as Hoplites, Pharaohs, and Longboats, respectively) but also mythological creatures such as Minotaurs, Centaurs, Phoenixes, and Valkyries in order to crush opponents.

System Requirements

- Microsoft Windows 98/Me/XP/2000
- PC with 450 MHz equivalent or higher processor
- 128 MB of system RAM
- 1.5 GB available hard disk space
- 32x speed or faster CD-ROM drive
- 16 MB video card required
- Sound card, speakers or headphones required for audio
- Microsoft Mouse or compatible pointing device
- 56.6 Kbps or better modem for online play

Multiplayer Features

- Dedicated Server
- DirectPlay
- Gamespy

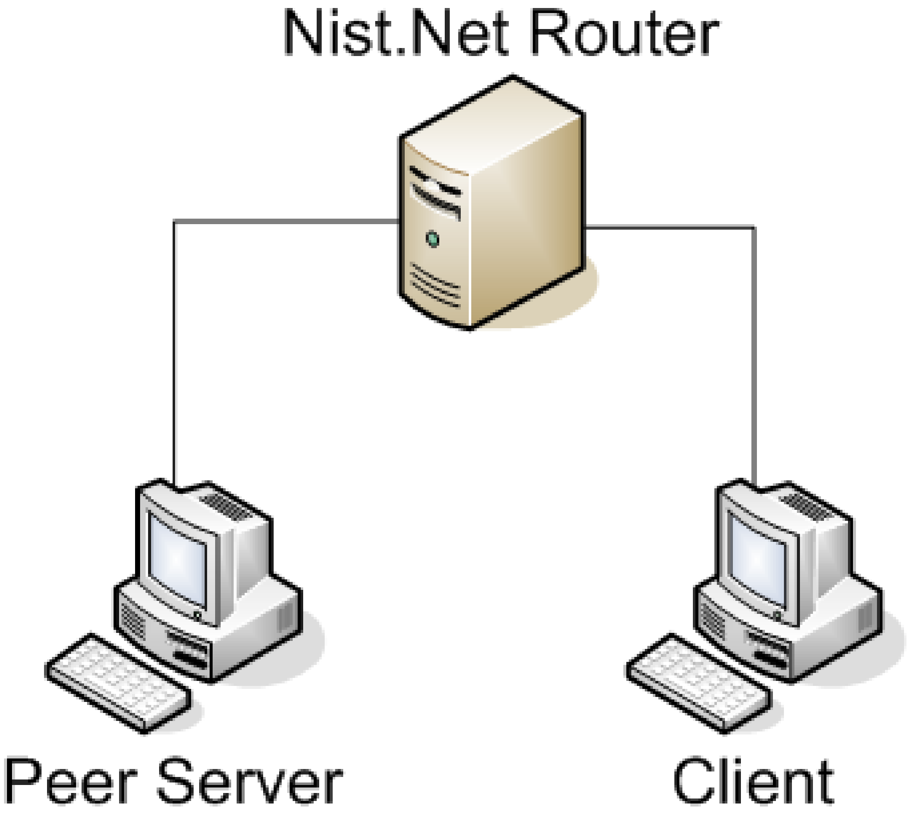


Figure 8.2: Nist.Net Network Diagram

8.5 Experiment

Participants will play a computer game under changing computer/computer network performance to gather information on their gaming satisfaction perception. Every half hour approximately the participants will be asked to fill in a brief survey while other players use the computers.

The Following Computers and Software were be used

8.5.1 Client Computers

- Celeron 1.7
- 512 MB RAM
- Geforece 440 128 MB
- 80 GB HDD
- Windows XP Pro

The client computer scored a 3dMark 2003 benchmark value of 173 3DMarks. 3dMark 2003 is a computer games bench-marking tool used to assigned a measurable value about the computer performance quality to compare against other computer setups.

8.5.2 Router Computer Specification

- Pentium-4 1.5
- 512 MB RAM
- Geforce 2 MX 400 64 MB
- 80 GB HDD
- Gentoo Linux with a 2.4.30 Linux Kernel

8.5.3 Software Used

- Nist.Net - Nistnet is used to limit the bandwidth between the 2 gaming machines [CS03]
- TCPDump - Used to record Network Traffic between the 2 gaming machines

8.6 Methodology

The experiment consisted of four people playing each other in various games and network conditions. The first session would be the establishment of a control group in which no wireless effect is applied. The subsequent sessions will progress downwards with worse and worse network conditions with the strategy of causing visible performance conditions. After each game session, the 2 players will fill up a survey form with a MoS type system with a 0 to 10 scale.

After the survey phase, the results were analyzed using regression analysis and were plotted in the form of response curve diagram. A Student t-test was applied to the different results to see if there is a link between different games and which coefficients are significance.

8.7 Questions

The following seven questions were used in the survey for the experiment. Question 1 was not asked during the first session as it is a baseline in which players compare their experiments from subsequent sessions with the first one.

Question 1 (Comparative Quality)

I really enjoyed the game and found no difference in game play from the first session

Do you
0 1 2 3 4 5 6 7 8 9 10
Strongly Disagree Strongly Agree
with the statement?

Question 2 (Quality Relative to Normal Game Play)

I really enjoyed the game and found no difference in game play from my normal gaming experience

Do you
0 1 2 3 4 5 6 7 8 9 10
Strongly Disagree Strongly Agree
with the statement?

Question 3 (Fairness)

I was not disadvantaged relative to my opponent

Do you
0 1 2 3 4 5 6 7 8 9 10
Strongly Disagree Strongly Agree
with the statement?

Question 4 (Responsiveness)

I feel that there was no lag compared to my normal gaming experience when I played the game during this session

Do you
0 1 2 3 4 5 6 7 8 9 10
Strongly Disagree Strongly Agree
with the statement?

Question 5 (Impact on Playing Style)

The System Setup and performance had no impact on the style of play I adopted during this session.

Do you
0 1 2 3 4 5 6 7 8 9 10
Strongly Disagree Strongly Agree
with the statement?

Question 6 (Impact on Playing Actions)

The System Setup and performance had no impact on the playing actions during this session.

Do you

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree										Strongly Agree

with the statement?

Question 7 (Enjoyment)

I really enjoyed the session

Do you

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree										Strongly Agree

with the statement?

8.8 Results

Based on the experiment, the following results and analysis of the study are presented here.

8.8.1 Age of Mythology Survey Response

Table 8.2 refers to the answers given by the participants of the experiment for the game Age of Mythology.

Conditions	**Client**							**Server**						
Metric	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**
1		10	10	10	10	10	10		10	10	10	10	10	10
0.744608310	8	8	10	7	10	10	10	10	8	9	4	10	10	10
0.575000000	5	8	8	7	10	10	10	9	8	9	4	9	9	8
0.0650646524	4	4	6	2	3	7	5	4	4	9	2	4	4	4
0.350000000	10	10	10	10	10	10	10	10	10	10	10	10	10	10

Table 8.2: Age of Mythology Survey Response

8.8.2 Battlefield 1942 Survey Response

Table 8.4 refers to the answers given by the participants of the experiment for the game Battlefield 1942.

8.8.3 Mechwarrior 4: Mercenaries Survey Response

Table 8.6 refers to the answers given by the participants of the experiment for the game Mechwarrior 4: Mercenaries.

Conditions	Client							Server						
Metric	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**
1		10	10	10	10	10	10		10	10	10	10	10	10
0.744608310	8	10	10	7	10	10	10	10	10	10	10	10	10	10
0.426514552	0	0	0	0	0	0	0	4	4	10	10	10	10	4
0.201514552	0	0	0	0	0	0	0	4	4	10	10	10	10	4
0.446127639	2	1	0	0	1	5	1	5	5	10	10	10	10	5
0.740064652	0	0	0	0	0	0	0	4	4	10	10	10	10	4
0.325000000	0	0	0	0	0	0	0	9	9	10	10	9	9	10
0.509930570	2	2	0	0	0	4	4	9	9	10	10	9	9	10
0.696127639	3	3	2	2	5	2	4	9	9	10	10	8	8	10
0.746127639	5	4	3	5	4	4	4	9	9	10	10	7	7	10
0.746127639	5	5	5	5	5	5	5	8	8	10	10	6	6	9
0.759677739	8	8	9	8	9	9	9	8	8	10	10	5	5	9
0.721134185	7	6	7	7	6	7	7	8	8	10	10	5	5	9

Table 8.4: Battlefield 1942 Survey Response

Conditions	Client							Server						
Metric	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**
1		10	10	10	10	10	10		10	10	10	10	10	10
0.44006465	2	2	2	2	2	2	2	6	5	2	4	10	10	5
0.184930570	0	0	0	0	0	0	0	5	5	0	4	5	5	4
0.359930570	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.478029105	7	7	7	7	7	7	7	7	6	5	7	7	7	6

Table 8.6: Mechwarrior 4: Mercenaries Survey Response

8.8.4 Regression Results for Player Response against Technical Quality

	Q2	Q3	Q4	Q5	Q6	Q7
Intercept	5.414	6.816	3.832	5.025	7.868	6.446
Intercept Standard Error	1.799	1.25	2.457	2.111	0.904	1.508
t Stat	3.008	5.452	1.559	2.38	8.695	4.274
Slop	4.727	3.625	6.157	6.535	2.8	4.668
Slop Standard Error	2.837	1.97	3.873	3.328	1.426	2.377
t Stat	1.665	1.839	1.589	1.963	1.963	1.963

Table 8.7: Regression Analysis on Age of Mythology Client

Table 8.7 shows the regression of client player response against technical quality in Age of Mythology.

	Q2	Q3	Q4	Q5	Q6	Q7
Intercept	5.414	9.128	3.35	5.565	5.565	5.395
Intercept Standard Error	1.799	0.528	3.377	1.714	1.714	1.738
t Stat	3.008	17.288	0.992	3.245	3.245	3.104
Slop	4.727	0.496	4.844	5.547	5.547	5.493
Slop Standard Error	2.837	0.832	5.323	2.703	2.703	2.74
t Stat	1.665	0.596	0.91	2.051	2.051	2.004

Table 8.8: Regression Analysis on Age of Mythology Server

Table 8.8 shows the regression of server player response against technical quality in Age of Mythology.

	Q2	Q3	Q4	Q5	Q6	Q7
Intercept	-4.444	-5.135	-5.011	-4.911	-3.105	-4.161
Intercept Standard Error	2.132	2.468	2.005	2.245	2.472	2.237
t Stat	-2.084	-2.08	-2.499	-2.187	-1.256	-1.86
Slop	13.242	13.984	13.537	14.119	11.953	13.406
Slop Standard Error	3.253	3.766	3.059	3.425	3.772	3.413
t Stat	4.07	3.712	4.424	4.121	3.168	3.927

Table 8.9: Regression Analysis on Battlefield 1942 Client

Table 8.9 shows the regression of client player response against technical quality in Battlefield 1942.

Table 8.10 shows the regression of server player response against technical quality in Battlefield 1942.

	Q2	Q3	Q4	Q5	Q6	Q7
Intercept	3.957	10	10	10.328	10.328	4.323
Intercept Standard Error	1.765	0	0	1.663	1.663	2.061
t Stat	2.241	DIV/0	DIV/0	6.209	6.209	2.097
Slop	5.649	0	0	-3.134	-3.134	5.928
Slop Standard Error	2.694	0	0	2.537	2.537	3.144
t Stat	2.096	DIV/0	DIV/0	-1.235	-1.235	1.885

Table 8.10: Regression Analysis on Battlefield 1942 Server

	Q2	Q3	Q4	Q5	Q6	Q7
Intercept	-2.632	-2.632	-2.632	-2.632	-2.632	-2.632
Intercept Standard Error	2.211	2.211	2.211	2.211	2.211	2.211
t Stat	-1.19	-1.19	-1.19	-1.19	-1.19	-1.19
Slop	13.059	13.059	13.059	13.059	13.059	13.059
Slop Standard Error	3.926	3.926	3.926	3.926	3.926	3.926
t Stat	3.325	3.325	3.325	3.325	3.325	3.325

Table 8.11: Regression Analysis on Mechwarrior 4: Mercenaries Client

Table 8.11 shows the regression of client player response against technical quality in Mechwarrior 4: Mercenaries.

	Q2	Q3	Q4	Q5	Q6	Q7
Intercept	0.989	-3.071	0.332	2.555	2.555	0.382
Intercept Standard Error	2.585	1.396	2.528	3.632	3.632	2.337
t Stat	0.382	-2.199	0.131	0.703	0.703	0.163
Slop	8.548	13.137	9.475	7.804	7.804	9.373
Slop Standard Error	4.591	2.479	4.489	6.449	6.449	4.149
t Stat	1.861	5.297	2.11	1.21	1.21	2.258

Table 8.12: Regression Analysis on Mechwarrior 4: Mercenaries Server

Table 8.12 shows the regression of server player response against technical quality in Mechwarrior 4: Mercenaries.

8.8.5 Student's t-Test

	Q2	Q3	Q4	Q5	Q6	Q7
Degrees of Freedom	4	4	4	4	4	4
T-Test on Intercept	0	1.703	0.115	0.198	1.187	0.456
t-Distribution	0.5	0.069	0.455	0.424	0.139	0.331
Degrees of Freedom	4	4	4	4	4	4
T-Test on Slope	0	1.462	0.199	0.23	0.898	0.227
t-Distribution	0.5	0.096	0.424	0.412	0.201	0.413

Table 8.13: Student's t-Test on Age of Mythology Client versus Age of Mythology Server

Table 8.13 shows Student's t-Test to the difference between slope and intercept coefficients on Age of Mythology Client and Age of Mythology Server

	Q2	Q3	Q4	Q5	Q6	Q7
Degrees of Freedom	4	4	4	4	4	4
T-Test on Intercept	3.034	6.13	7.485	5.454	4.508	2.789
t-Distribution	0.003	1.79862E-06	8.74398E-08	8.83836E-06	8.71972E-05	0.005
Degrees of Freedom	4	4	4	4	4	4
T-Test on Slope	1.797	3.712	4.424	4.047	3.318	1.611
t-Distribution	0.043	0.000606113	0.000107017	0.000268827	0.001	0.06

Table 8.14: Student's t-Test on Battlefield 1942 Client versus Battlefield 1942 Server

Table 8.14 shows Student's t-Test to the difference between slope and intercept coefficients on Battlefield 1942 Client and Battlefield 1942 Server

	Q2	Q3	Q4	Q5	Q6	Q7
Degrees of Freedom	4	4	4	4	4	4
T-Test on Intercept	1.064	0.167	0.882	1.22	1.22	0.937
t-Distribution	0.164	0.436	0.205	0.134	0.134	0.192
Degrees of Freedom	4	4	4	4	4	4
T-Test on Slope	0.746	0.016	0.6	0.696	0.696	0.645
t-Distribution	0.241	0.493	0.284	0.256	0.256	0.271

Table 8.15: Student's t-Test on Mechwarrior 4: Mercenaries Client versus Mechwarrior 4: Mercenaries Server

Table 8.15 shows Student's t-Test to the difference between slope and intercept coefficients on Mechwarrior 4: Mercenaries Client and Mechwarrior 4: Mercenaries Server

	Q2	Q3	Q4	Q5	Q6	Q7
Degrees of Freedom	6	6	6	6	6	6
T-Test on Intercept	3.532	4.319	2.788	3.223	4.167	3.931
t-Distribution	0.002	0.000498878	0.008	0.003	0.000652492	0.000996207
Degrees of Freedom	6	6	6	6	6	6
T-Test on Slope	1.972	2.436	1.495	1.587	2.269	2.1
t-Distribution	0.036	0.015	0.08	0.069	0.021	0.028

Table 8.16: Student's t-Test on Age of Mythology Client versus Battlefield 1942 Client

	Q2	Q3	Q4	Q5	Q6	Q7
Degrees of Freedom	6	6	6	6	6	6
T-Test on Intercept	0.589	0.754	0.796	0.722	0.142	0.485
t-Distribution	0.283	0.232	0.22	0.241	0.444	0.317
Degrees of Freedom	6	6	6	6	6	6
T-Test on Slope	0.035	0.17	0.096	0.203	0.203	0.066
t-Distribution	0.485	0.433	0.462	0.421	0.421	0.473

Table 8.17: Student's t-Test on Battlefield 1942 Client versus Mechwarrior 4: Mercenaries Client

Table 8.16 shows Student's t-Test to the difference between slope and intercept coefficients on Age of Mythology Client and Battlefield 1942 Client

Table 8.17 shows Student's t-Test to the difference between slope and intercept coefficients on Battlefield 1942 Client and Mechwarrior 4: Mercenaries Client

	Q2	Q3	Q4	Q5	Q6	Q7
Degrees of Freedom	6	6	6	6	6	6
T-Test on Intercept	2.822	3.719	1.955	2.504	4.394	3.392
t-Distribution	0.023	0.01	0.061	0.033	0.005	0.013
Degrees of Freedom	6	6	6	6	6	6
T-Test on Slope	1.719	2.147	1.251	1.267	2.455	1.828
t-Distribution	0.08	0.049	0.139	0.136	0.035	0.07

Table 8.18: Student's t-Test on Mechwarrior 4: Mercenaries Client versus Age of Mythology Client

Table 8.18 shows Student's t-Test to the difference between slope and intercept coefficients on Mechwarrior 4: Mercenaries and Age of Mythology Client

Table 8.19 shows Student's t-Test to the difference between slope and intercept coefficients on Age of Mythology Server and Battlefield 1942 Server

Table 8.20 shows Student's t-Test to the difference between slope and intercept coefficients on Battlefield 1942 Server and Mechwarrior 4: Mercenaries Server

	Q2	Q3	Q4	Q5	Q6	Q7
Degrees of Freedom	6	6	6	6	6	6
T-Test on Intercept	0.577	1.65	1.969	1.993	1.993	0.397
t-Distribution	0.287	0.062	0.036	0.034	0.034	0.348
Degrees of Freedom	6	6	6	6	6	6
T-Test on Slope	0.235	0.596	0.91	2.341	2.341	0.104
t-Distribution	0.408	0.281	0.19	0.018	0.018	0.459

Table 8.19: Student's t-Test on Age of Mythology Server versus Battlefield 1942 Server

	Q2	Q3	Q4	Q5	Q6	Q7
Degrees of Freedom	6	6	6	6	6	6
T-Test on Intercept	0.947	9.359	3.823	1.945	1.945	1.264
t-Distribution	0.18	3.64346E-07	0.001	0.037	0.037	0.115
Degrees of Freedom	6	6	6	6	6	6
T-Test on Slope	0.544	5.297	2.11	1.578	1.578	0.661
t-Distribution	0.298	9.45013E-05	0.028	0.07	0.07	0.26

Table 8.20: Student's t-Test on Battlefield 1942 Server versus Mechwarrior 4: Mercenaries Server

Table 8.21 shows Student's t-Test to the difference between slope and intercept coefficients on Mechwarrior 4: Mercenaries Server and Age of Mythology Server

	Q2	Q3	Q4	Q5	Q6	Q7
Degrees of Freedom	6	6	6	6	6	6
T-Test on Intercept	1.404	8.171	0.715	0.749	0.749	1.721
t-Distribution	0.116	0.000610721	0.256	0.247	0.247	0.08
Degrees of Freedom	6	6	6	6	6	6
T-Test on Slope	1.861	5.297	2.11	1.21	1.21	2.258
t-Distribution	0.068	0.003	0.051	0.146	0.146	0.043

Table 8.21: Student’s t-Test on Mechwarrior 4: Mercenaries Server versus Age of Mythology Server

8.8.6 Response Curves

Age of Mythology Question 2

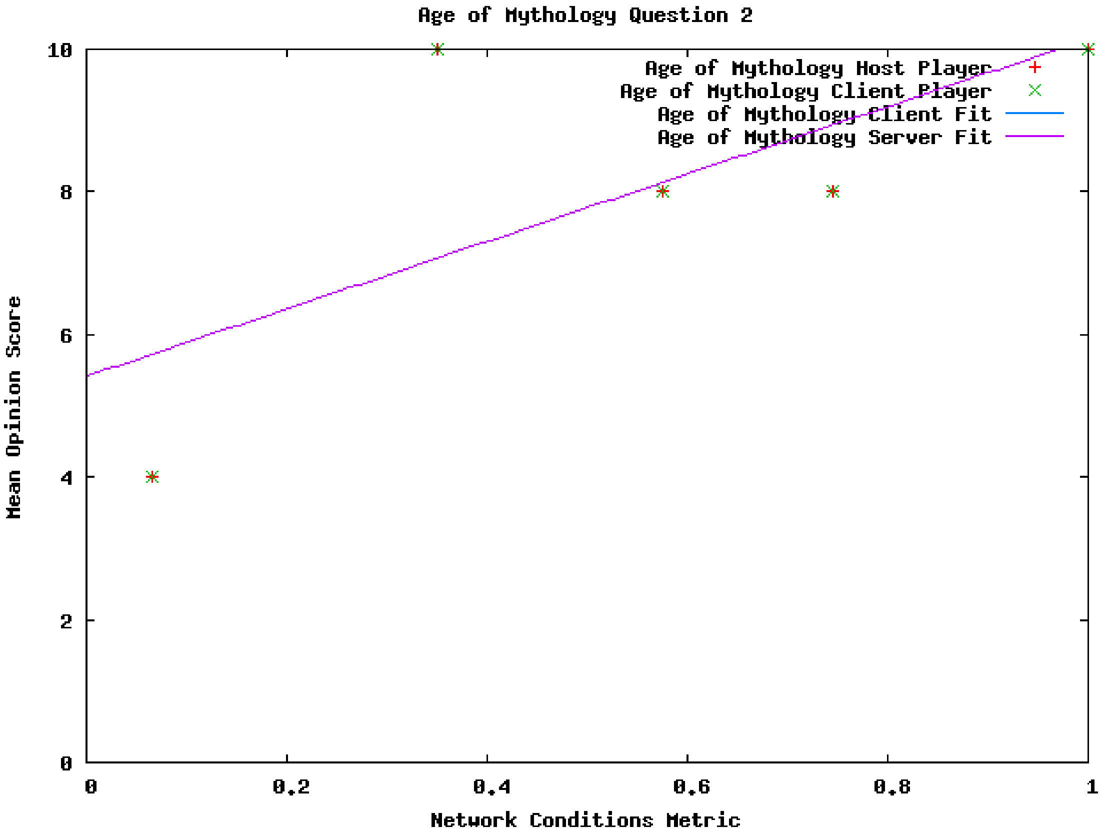


Figure 8.3: Age of Mythology Question 2

Figure 8.3 shows the estimated response curves for Question 2 in the Game Age of Mythology for both client and server.

Age of Mythology Question 3

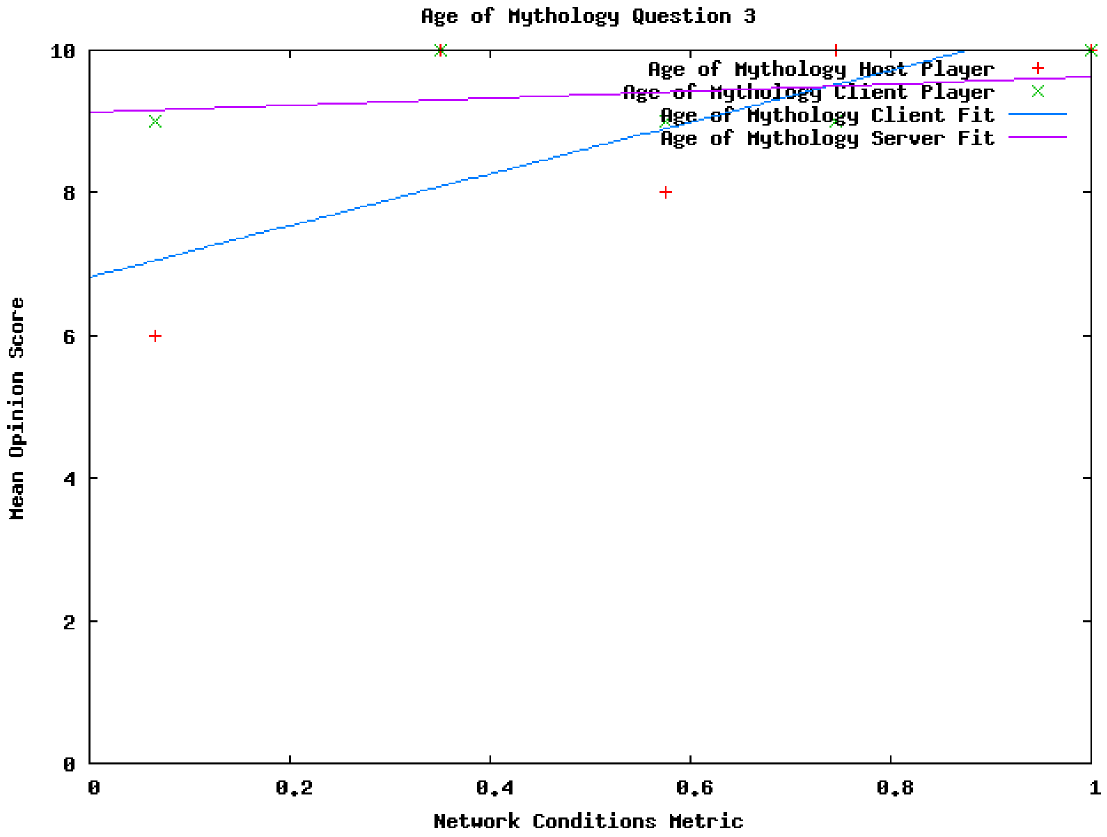


Figure 8.4: Age of Mythology Question 3

Figure 8.4 shows the estimated response curves for Question 3 in the Game Age of Mythology for both client and server.

Age of Mythology Question 4

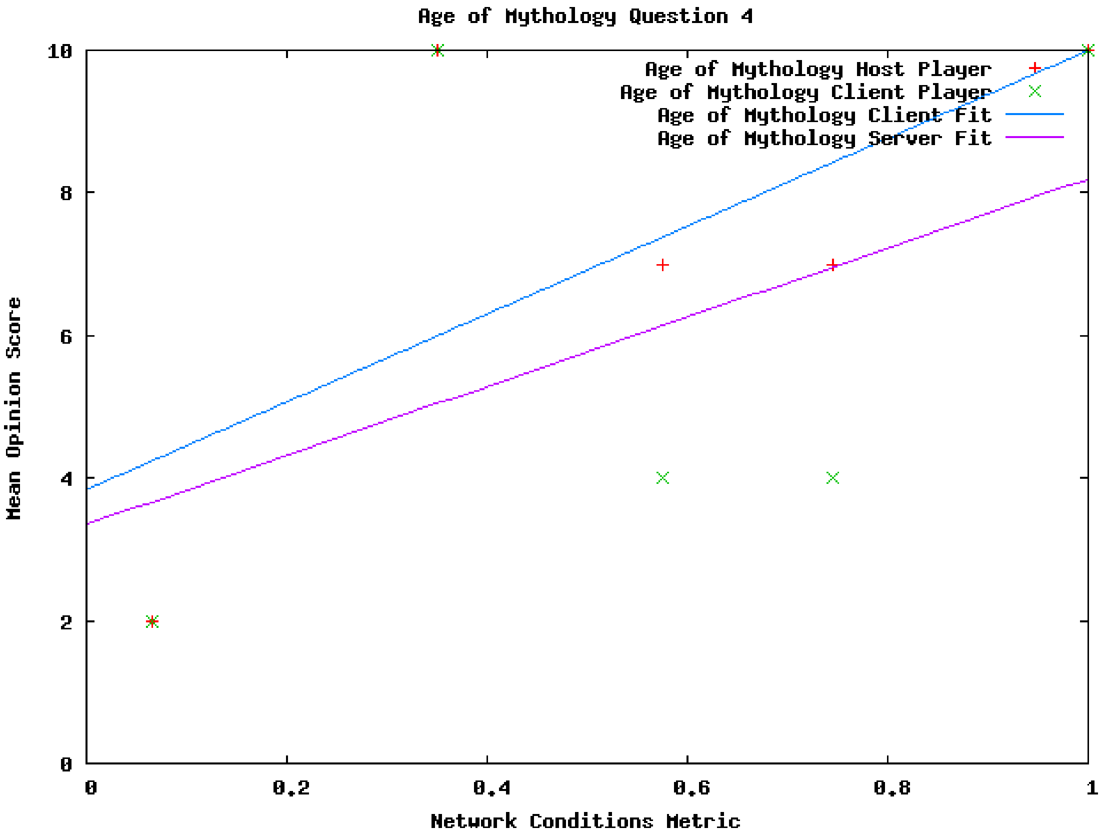


Figure 8.5: Age of Mythology Question 4

Figure 8.5 shows the estimated response curves for Question 4 in the Game Age of Mythology for both client and server.

Age of Mythology Question 5

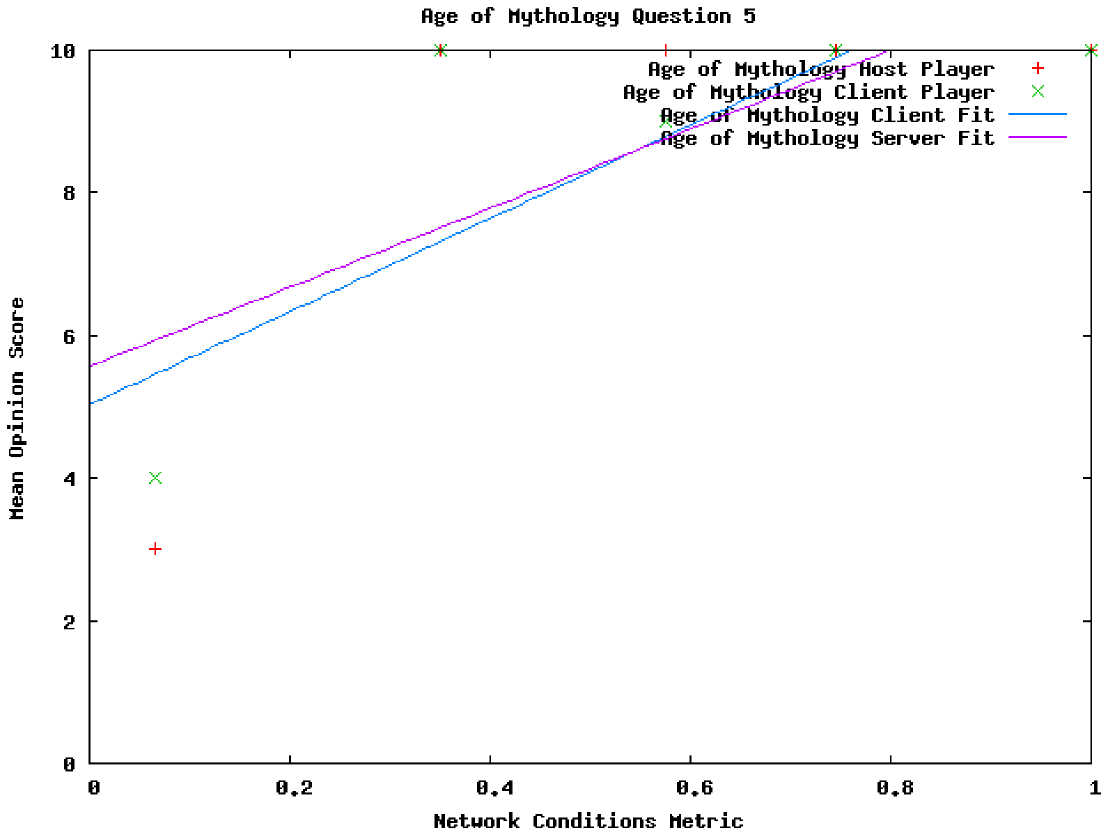


Figure 8.6: Age of Mythology Question 5

Figure 8.6 shows the estimated response curves for Question 5 in the Game Age of Mythology for both client and server.

Age of Mythology Question 6

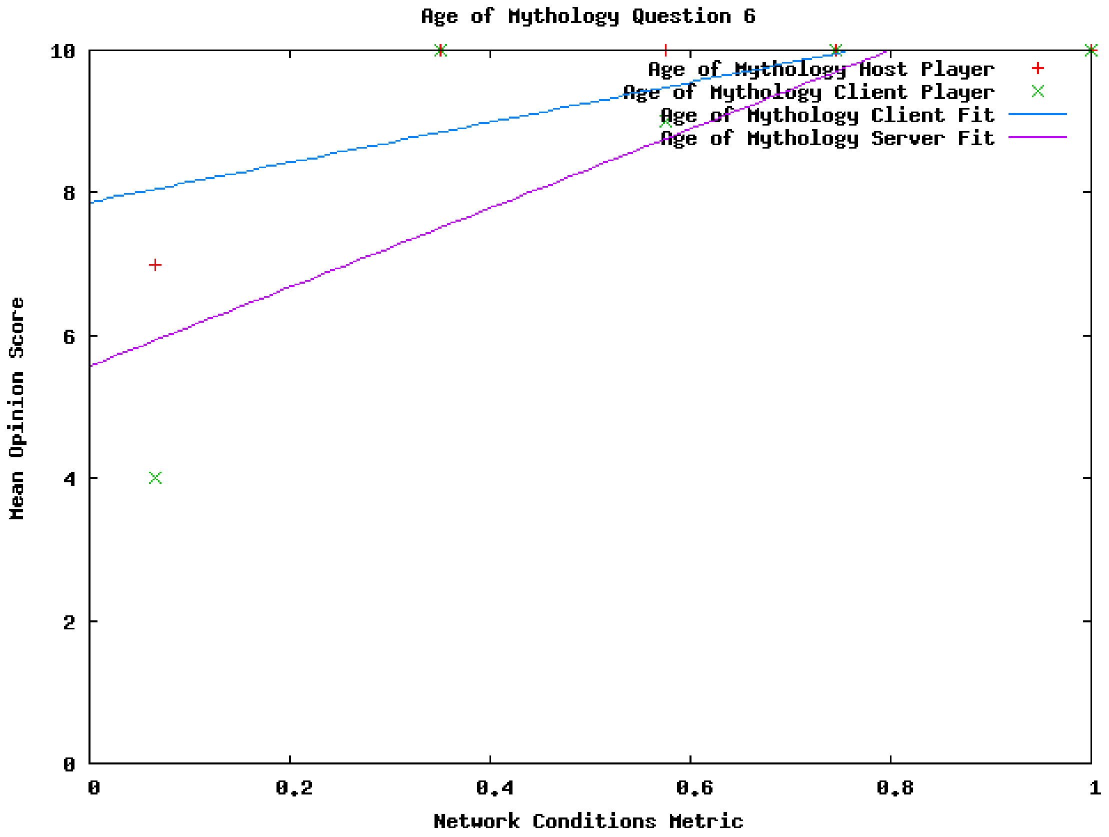


Figure 8.7: Age of Mythology Question 6

Figure 8.7 shows the estimated response curves for Question 6 in the Game Age of Mythology for both client and server.

Age of Mythology Question 7

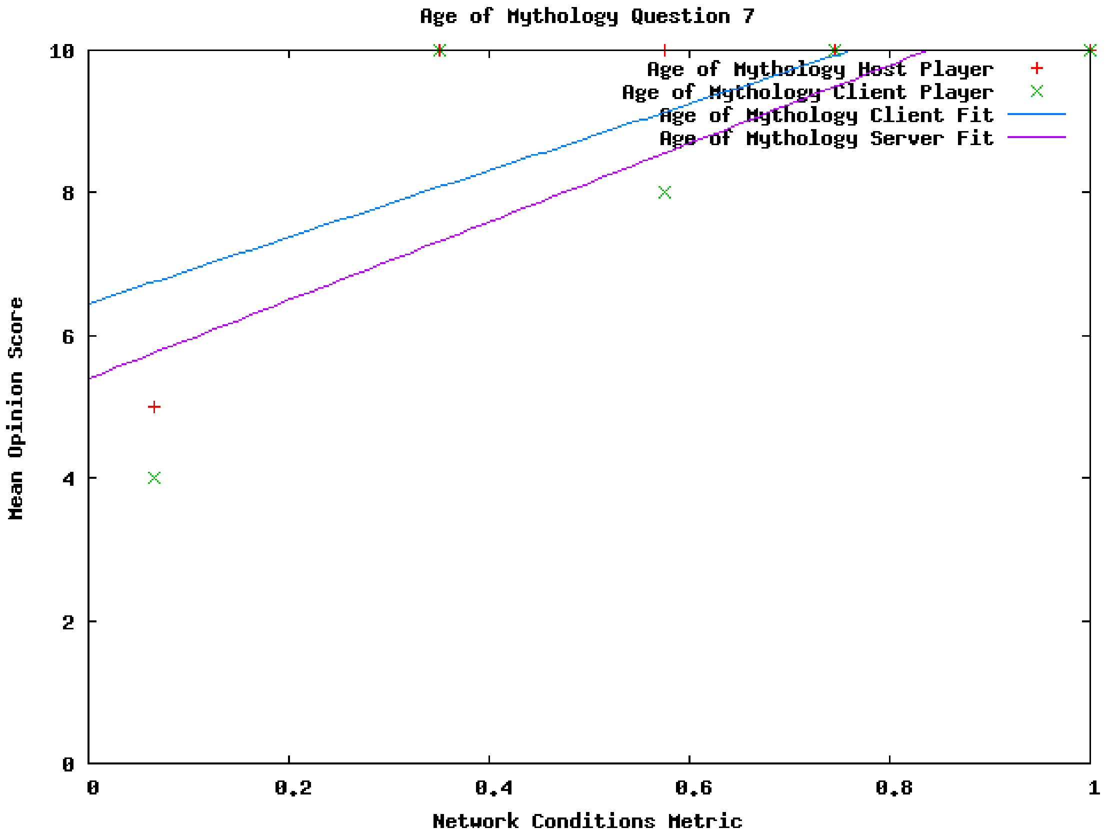


Figure 8.8: Age of Mythology Question 7

Figure 8.8 shows the estimated response curves for Question 7 in the Game Age of Mythology for both client and server.

Battlefield 1942 Question 2

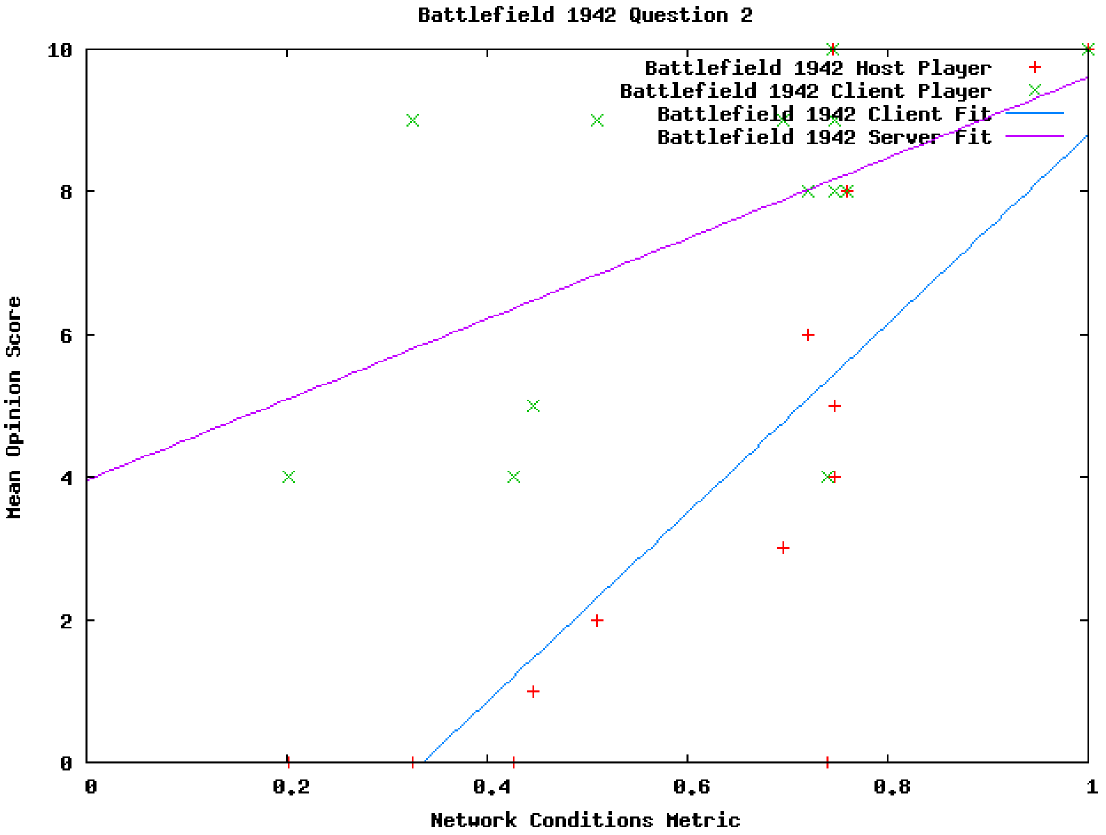


Figure 8.9: Battlefield 1942 Question 2

Figure 8.9 shows the estimated response curves for Question 2 in the Game Battlefield 1942 for both client and server.

Battlefield 1942 Question 3

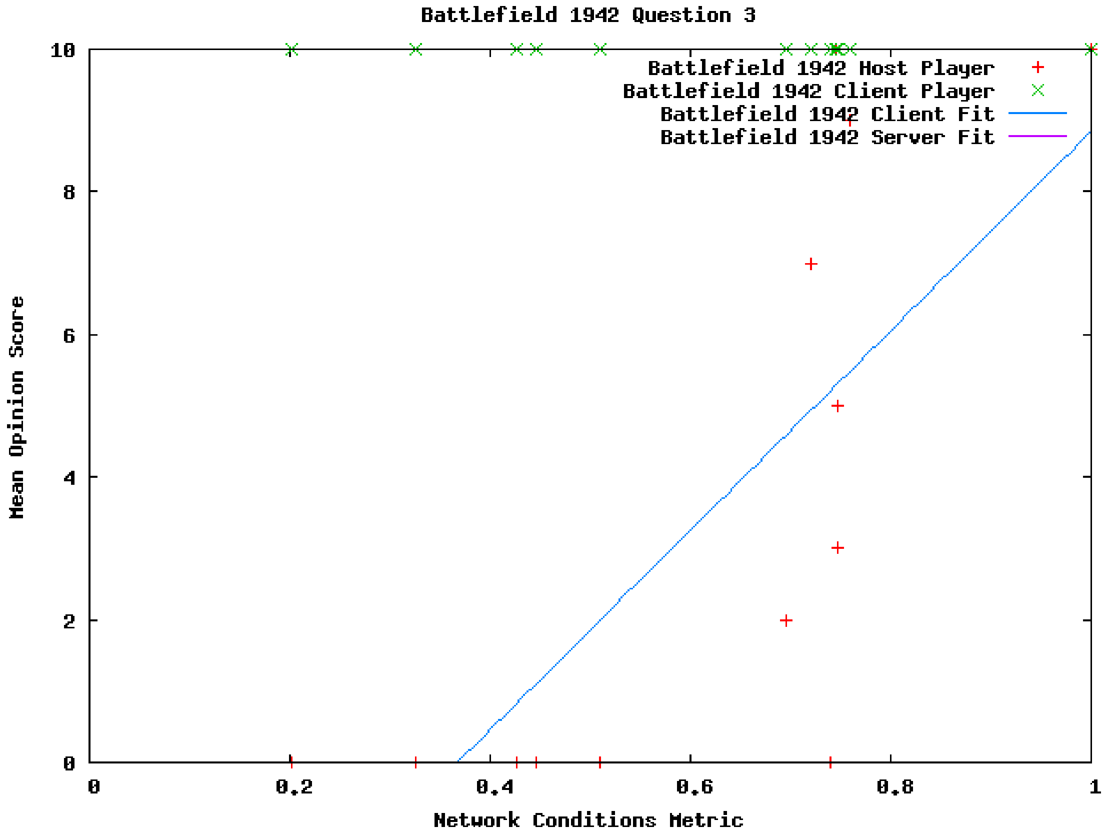


Figure 8.10: Battlefield 1942 Question 3

Figure 8.10 shows the estimated response curves for Question 3 in the Game Battlefield 1942 for both client and server.

Battlefield 1942 Question 4

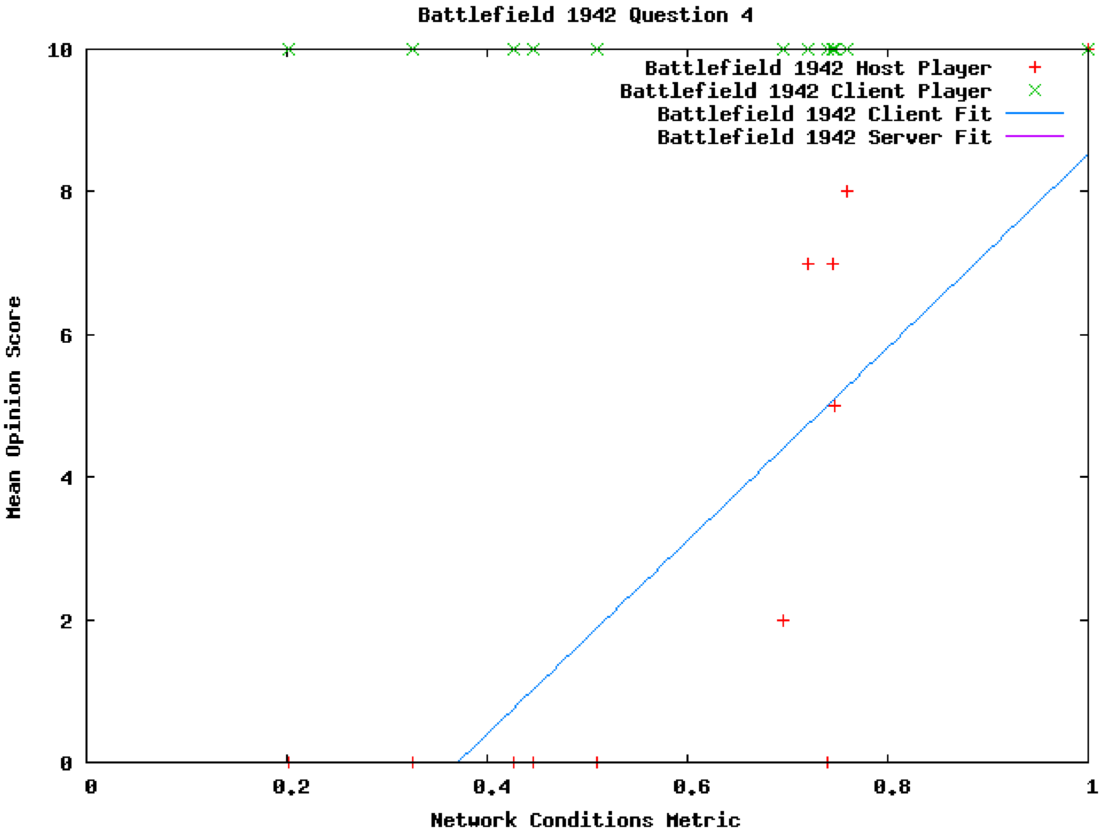


Figure 8.11: Battlefield 1942 Question 4

Figure 8.11 shows the estimated response curves for Question 4 in the Game Battlefield 1942 for both client and server.

Battlefield 1942 Question 5

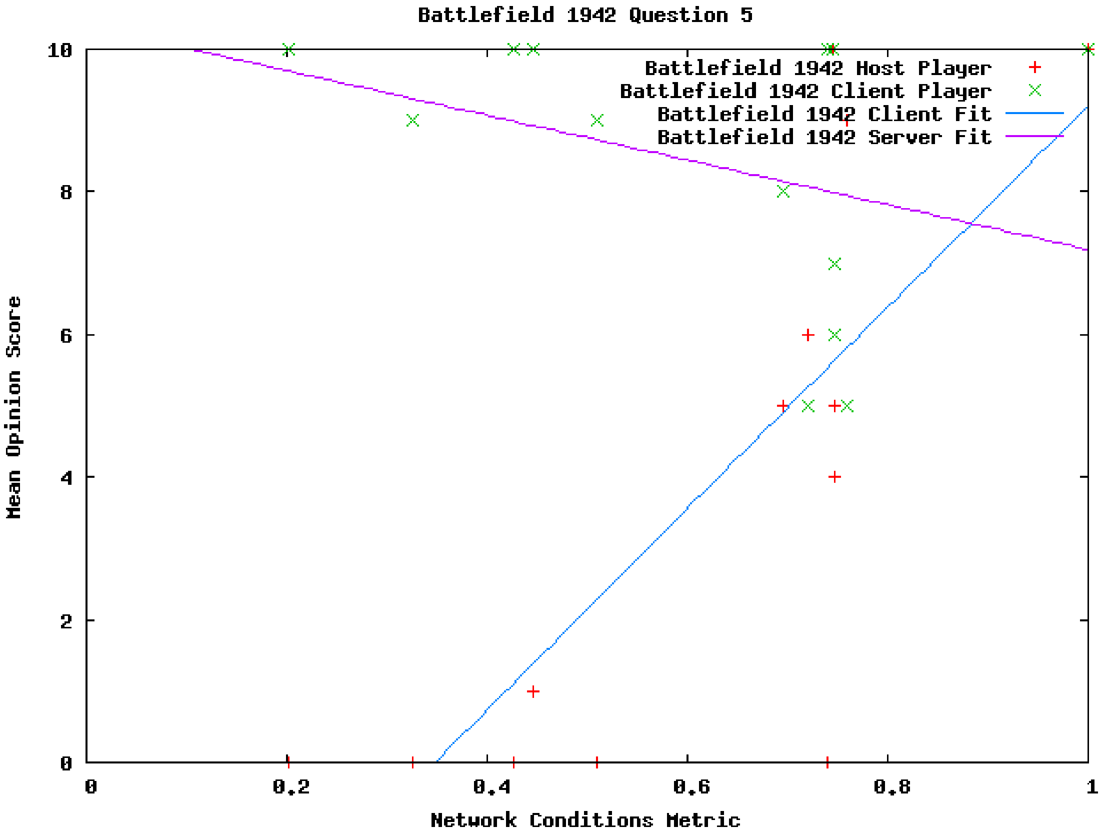


Figure 8.12: Battlefield 1942 Question 5

Figure 8.12 shows the estimated response curves for Question 5 in the Game Battlefield 1942 for both client and server.

Battlefield 1942 Question 6

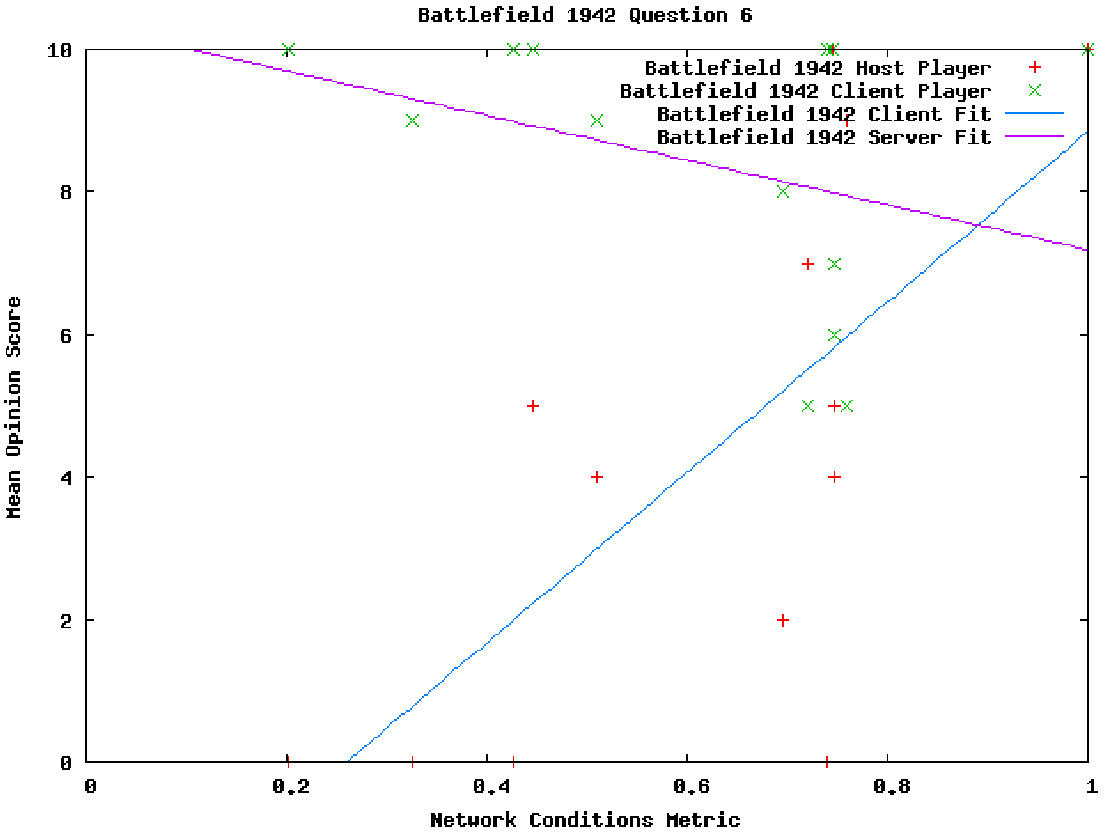


Figure 8.13: Battlefield 1942 Question 6

Figure 8.13 shows the estimated response curves for Question 6 in the Game Battlefield 1942 for both client and server.

Battlefield 1942 Question 7

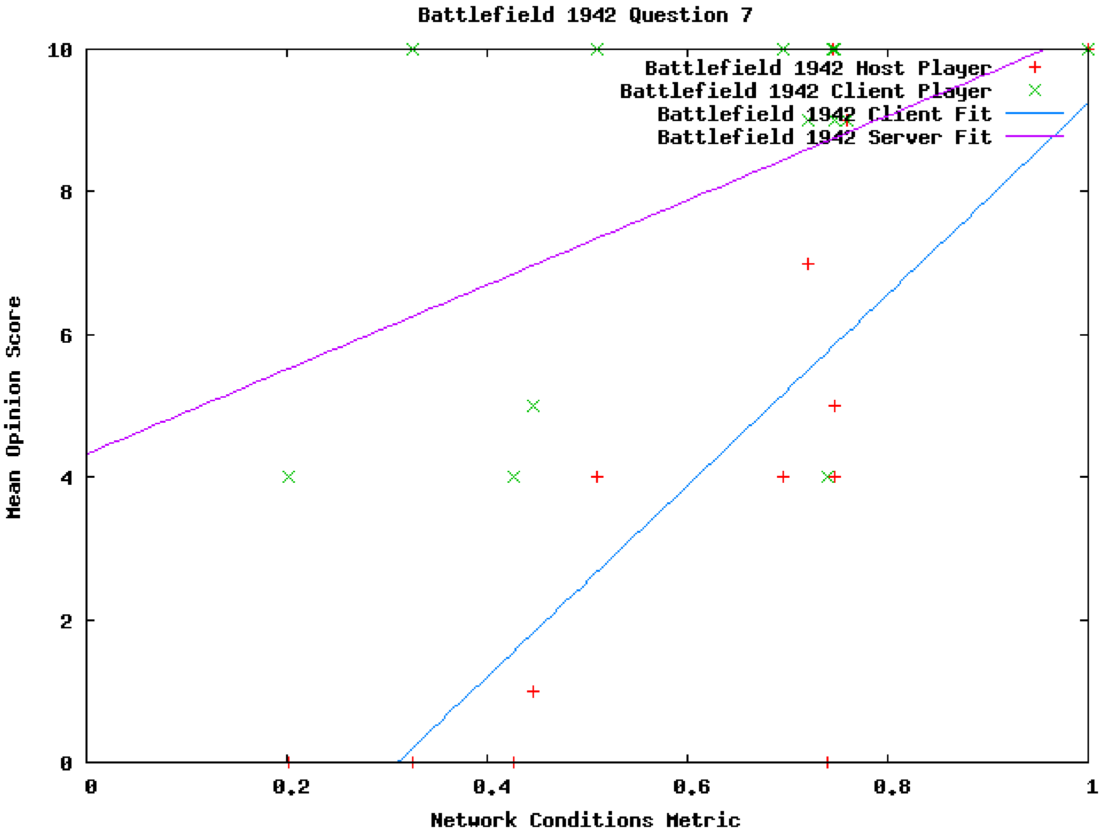


Figure 8.14: Battlefield 1942 Question 7

Figure 8.14 shows the estimated response curves for Question 7 in the Game Battlefield 1942 for both client and server.

Mechwarrior 4:Mercenaries Question 2

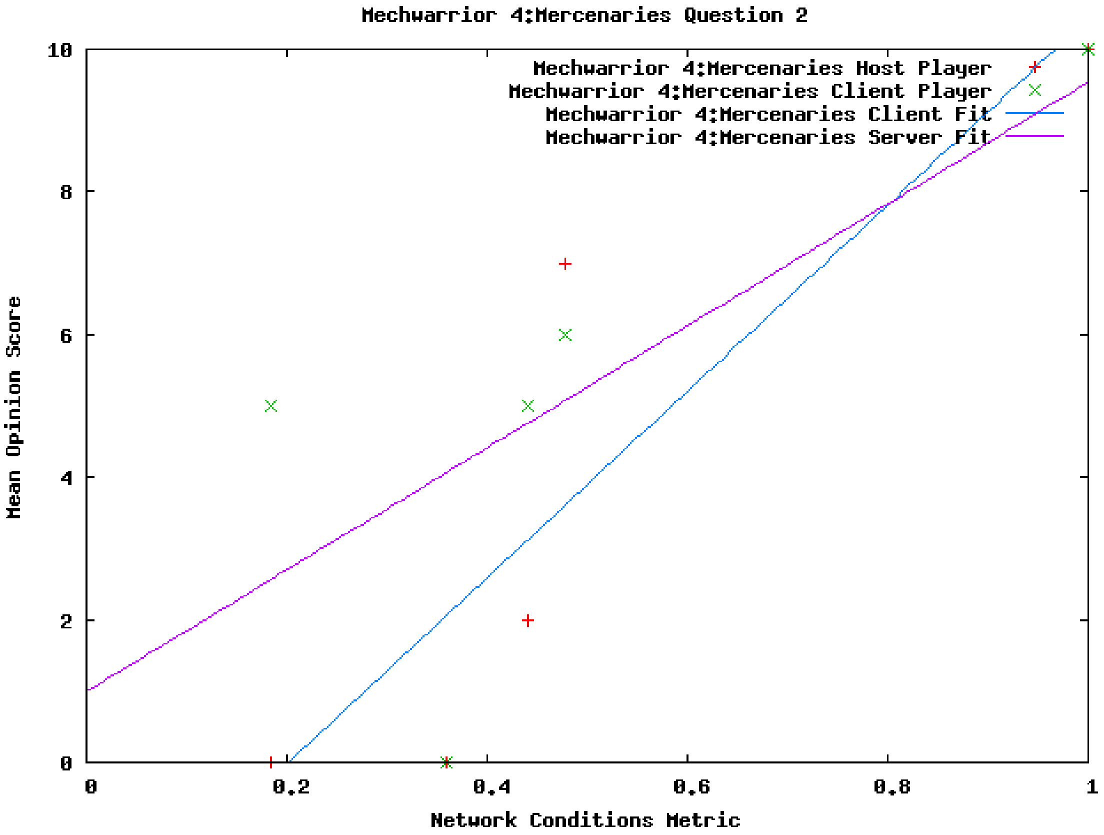


Figure 8.15: Mechwarrior 4:Mercenaries Question 2

Figure 8.15 shows the estimated response curves for Question 2 in the Game Mechwarrior:4 Mercenaries for both client and server.

Mechwarrior 4:Mercenaries Question 3

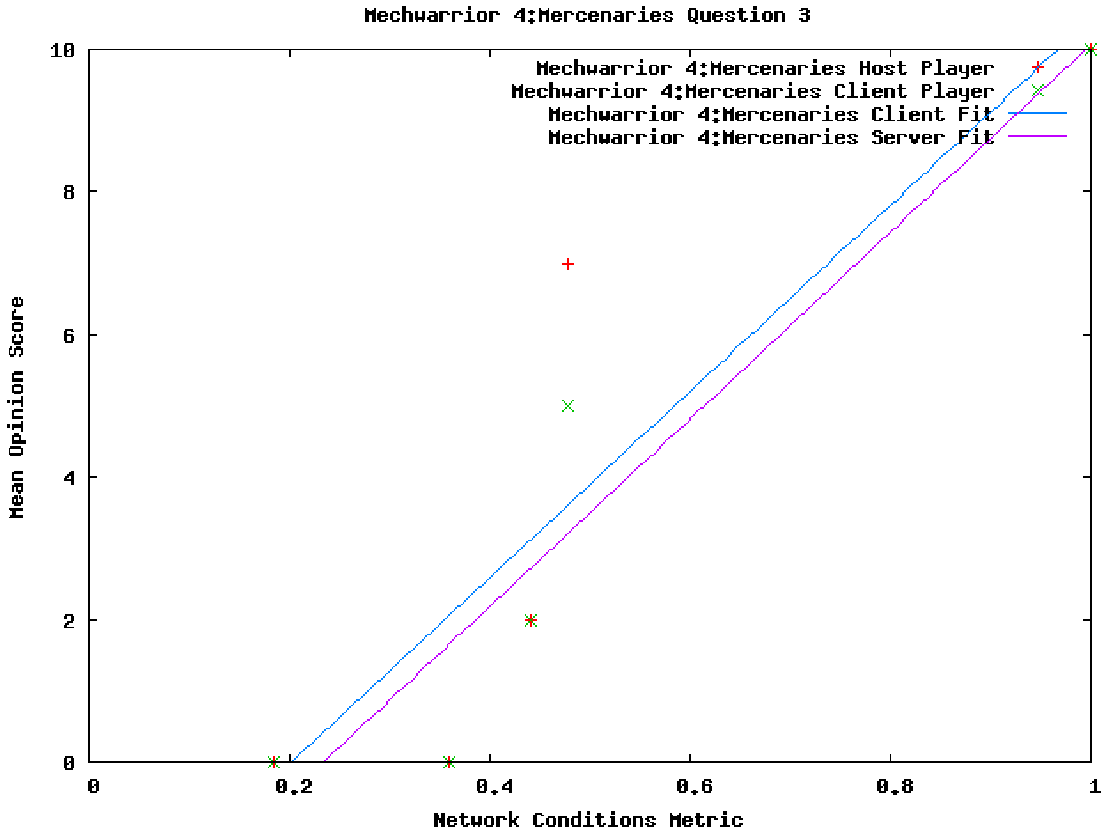


Figure 8.16: Mechwarrior 4:Mercenaries Question 3

Figure 8.16 shows the estimated response curves for Question 3 in the Game Mechwarrior:4 Mercenaries for both client and server.

Mechwarrior 4:Mercenaries Question 4

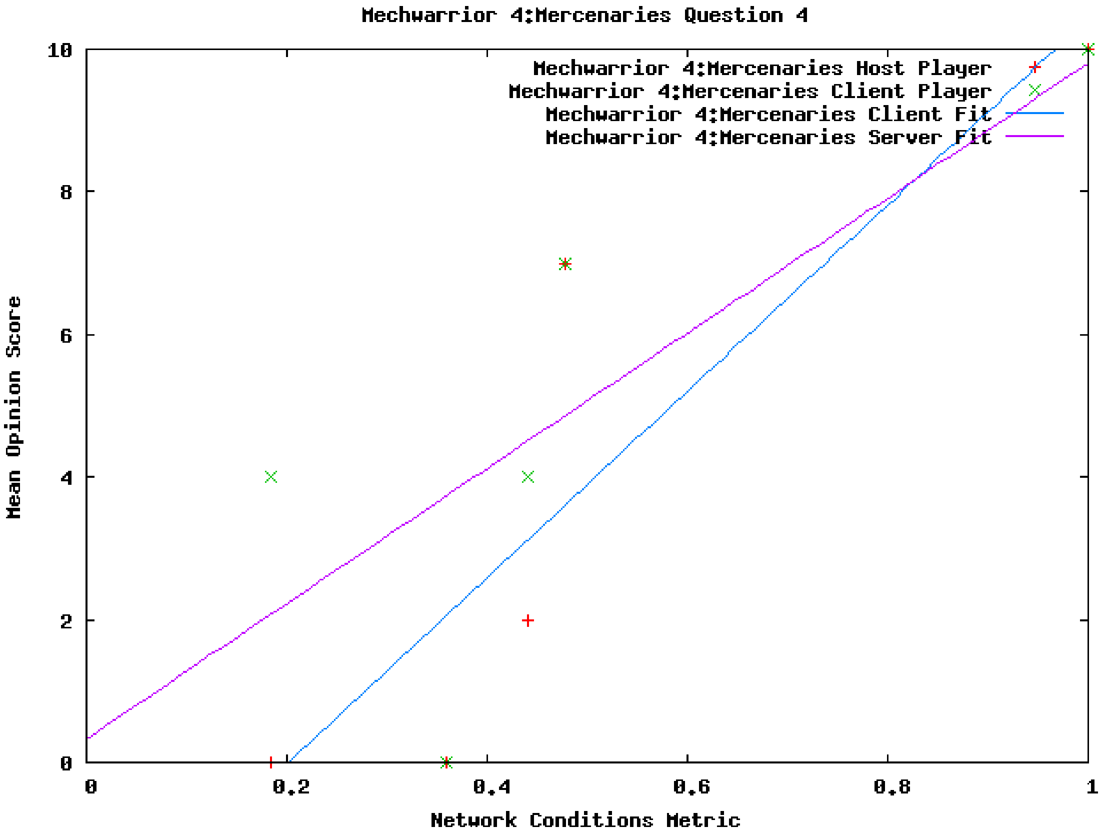


Figure 8.17: Mechwarrior 4:Mercenaries Question 4

Figure 8.17 shows the estimated response curves for Question 4 in the Game Mechwarrior:4 Mercenaries for both client and server.

Mechwarrior 4:Mercenaries Question 5

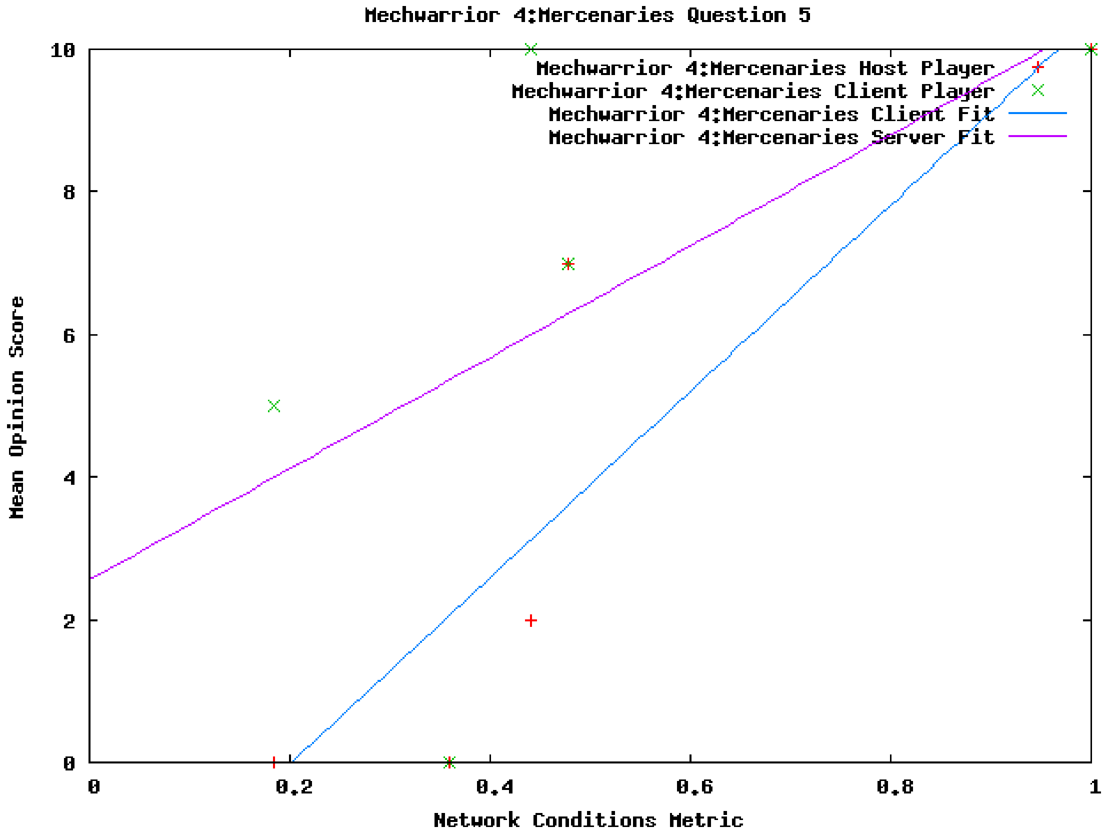


Figure 8.18: Mechwarrior 4:Mercenaries Question 5

Figure 8.18 shows the estimated response curves for Question 5 in the Game Mechwarrior:4 Mercenaries for both client and server.

Mechwarrior 4:Mercenaries Question 6

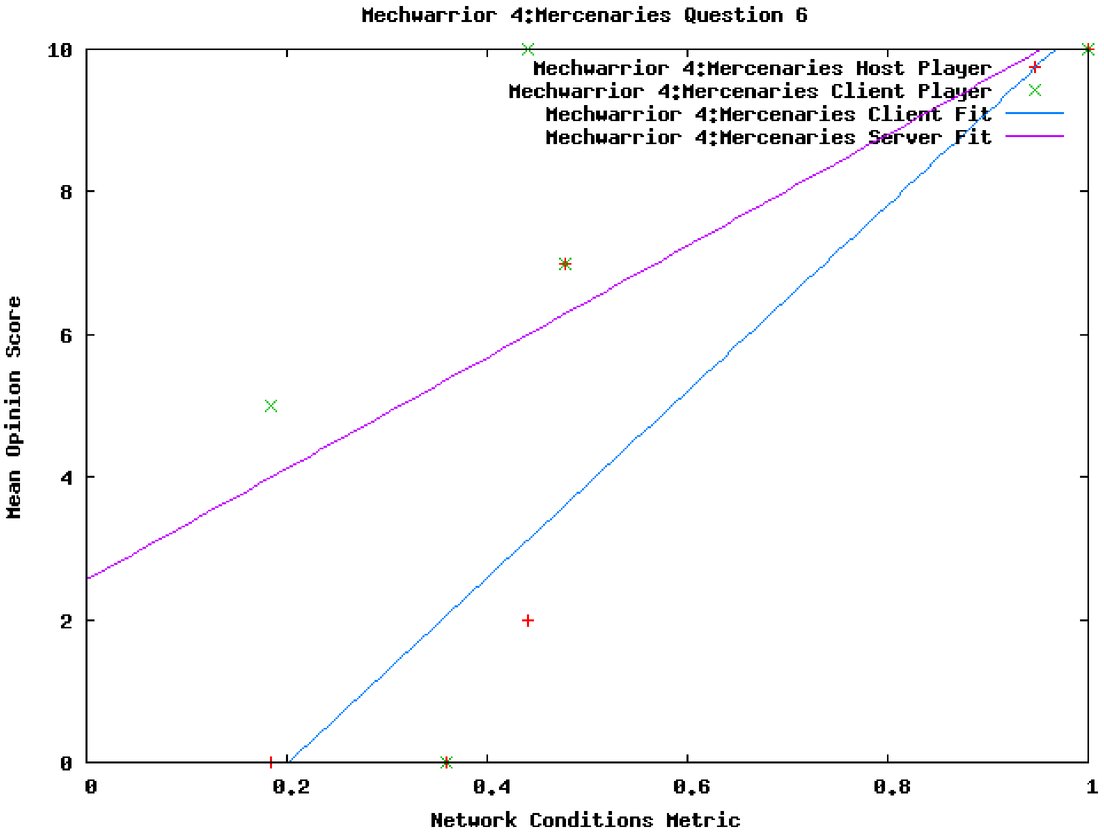


Figure 8.19: Mechwarrior 4:Mercenaries Question 6

Figure 8.19 shows the estimated response curves for Question 6 in the Game Mechwarrior:4 Mercenaries for both client and server.

Mechwarrior 4:Mercenaries Question 7

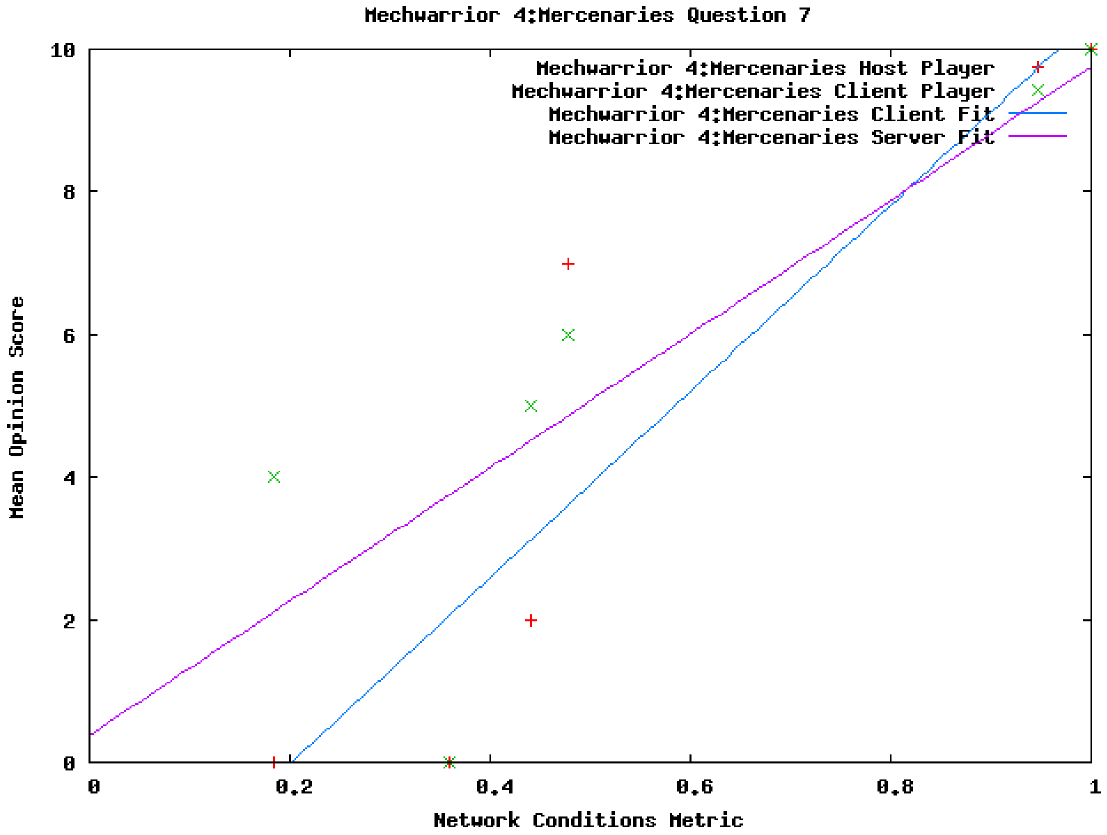


Figure 8.20: Mechwarrior 4:Mercenaries Question 7

Figure 8.20 shows the estimated response curves for Question 7 in the Game Mechwarrior:4 Mercenaries for both client and server.

Client Question 2

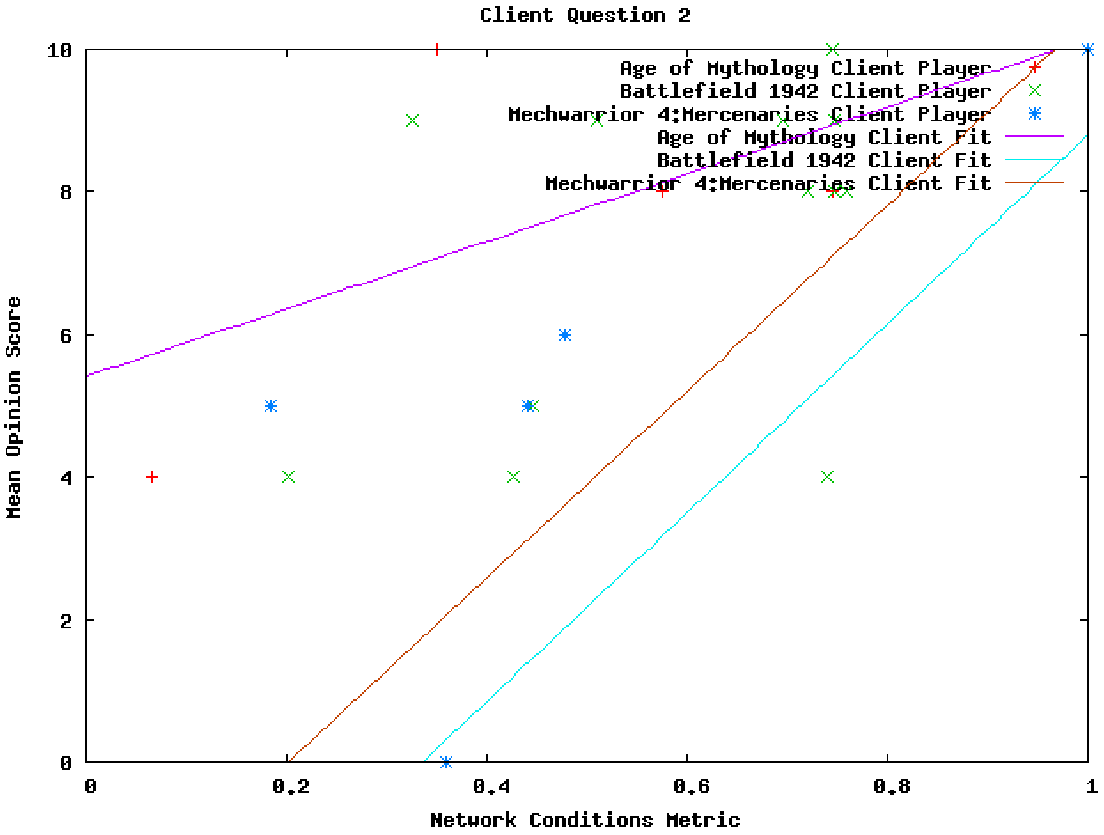


Figure 8.21: Client Question 2

Figure 8.21 shows the estimated response curves for Question 2 in the all 3 games for the client.

Client Question 3

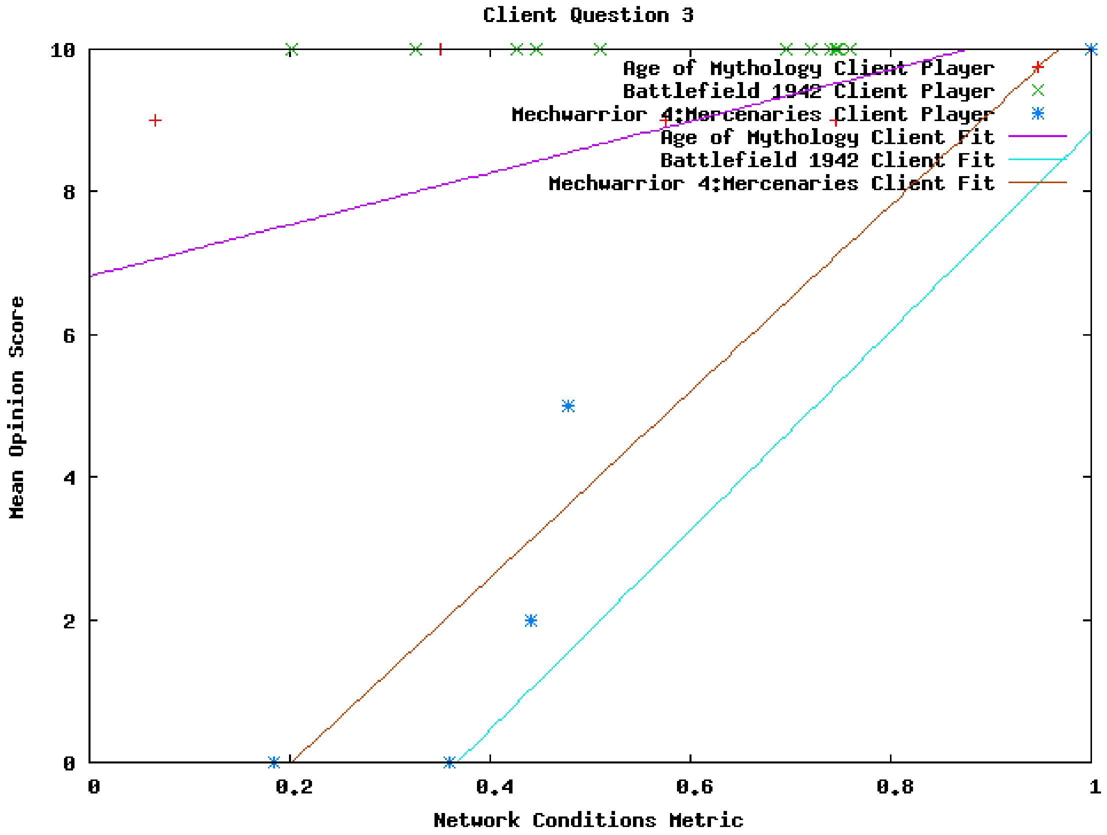


Figure 8.22: Client Question 3

Figure 8.22 shows the estimated response curves for Question 3 in the all 3 games for the client.

Client Question 4

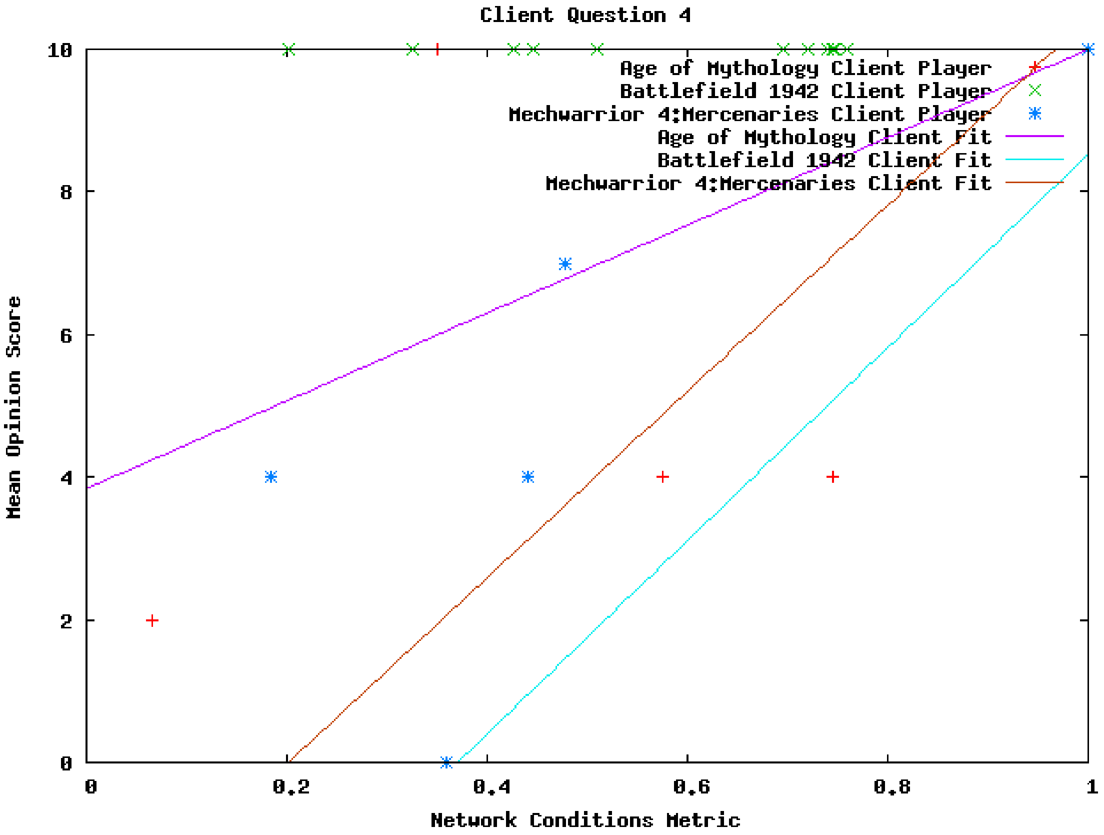


Figure 8.23: Client Question 4

Figure 8.23 shows the estimated response curves for Question 4 in the all 3 games for the client.

Client Question 5

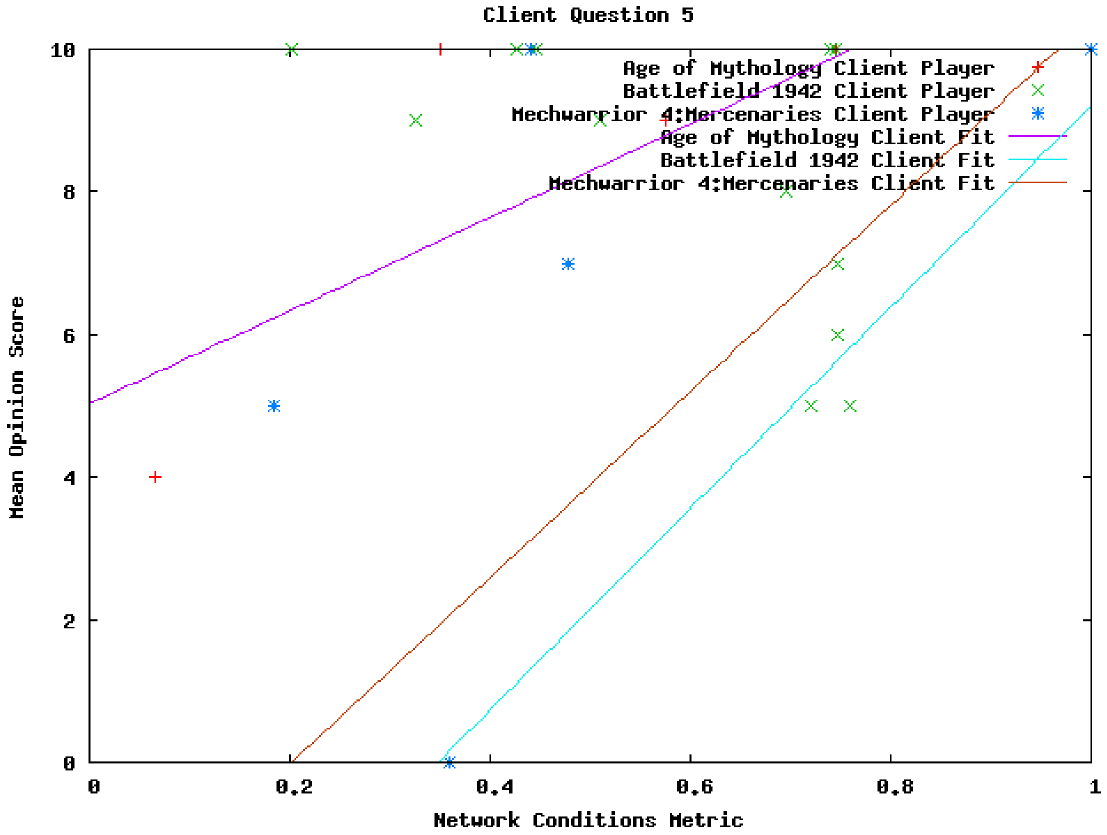


Figure 8.24: Client Question 5

Figure 8.24 shows the estimated response curves for Question 5 in the all 3 games for the client.

Client Question 6

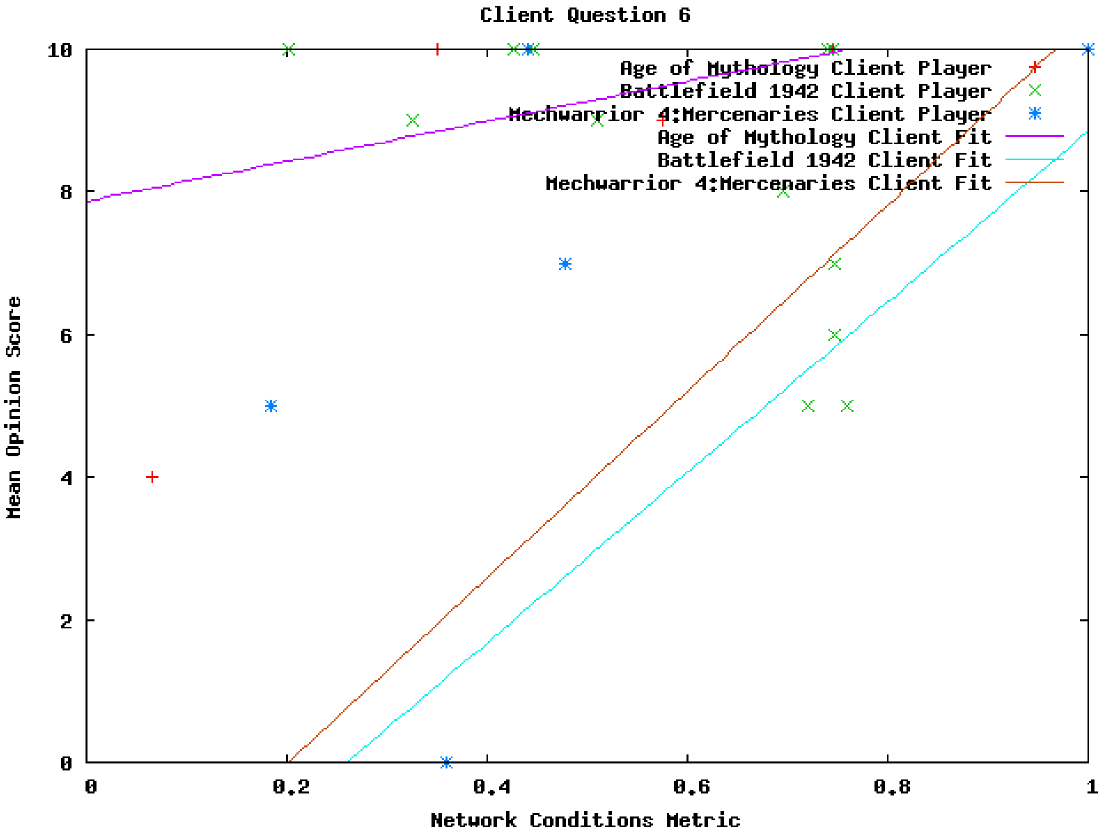


Figure 8.25: Client Question 6

Figure 8.25 shows the estimated response curves for Question 6 in the all 3 games for the client.

Client Question 7

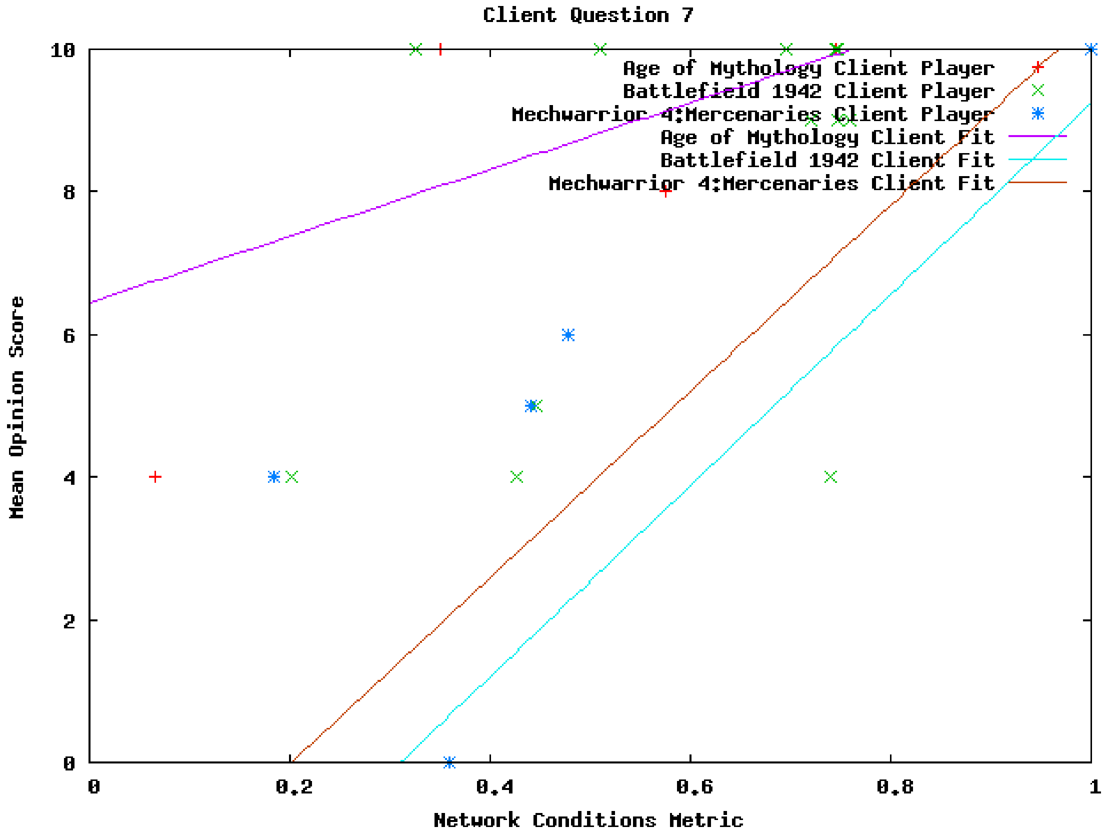


Figure 8.26: Client Question 7

Figure 8.26 shows the estimated response curves for Question 7 in the all 3 games for the client.

Server Question 2

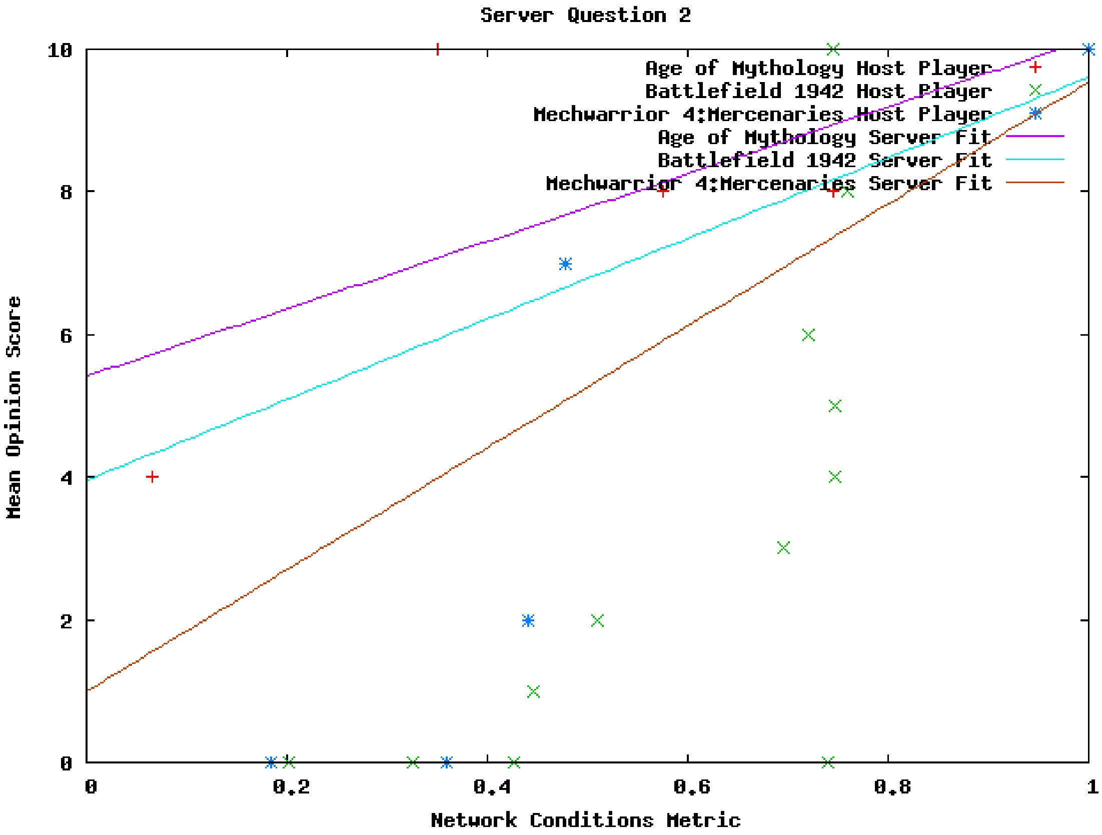


Figure 8.27: Server Question 2

Figure 8.27 shows the estimated response curves for Question 2 in the all 3 games for the server.

Server Question 3

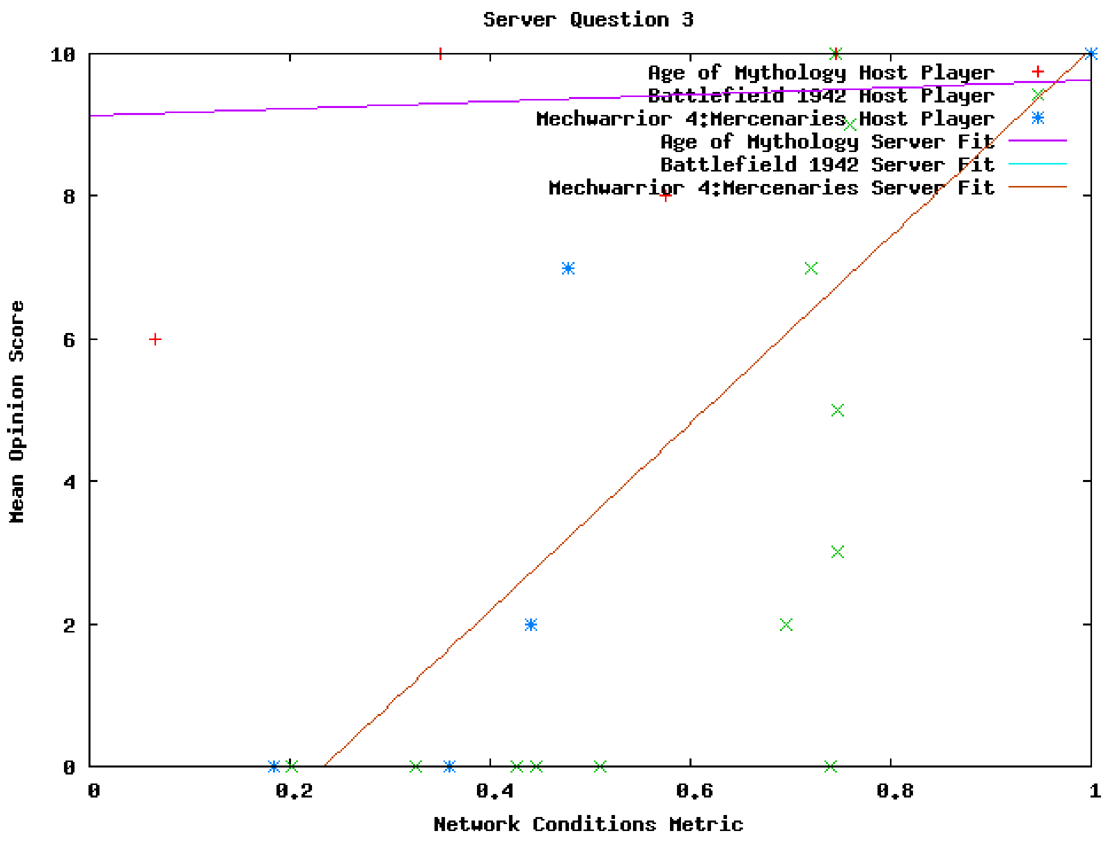


Figure 8.28: Server Question 3

Figure 8.28 shows the estimated response curves for Question 3 in the all 3 games for the server.

Server Question 4

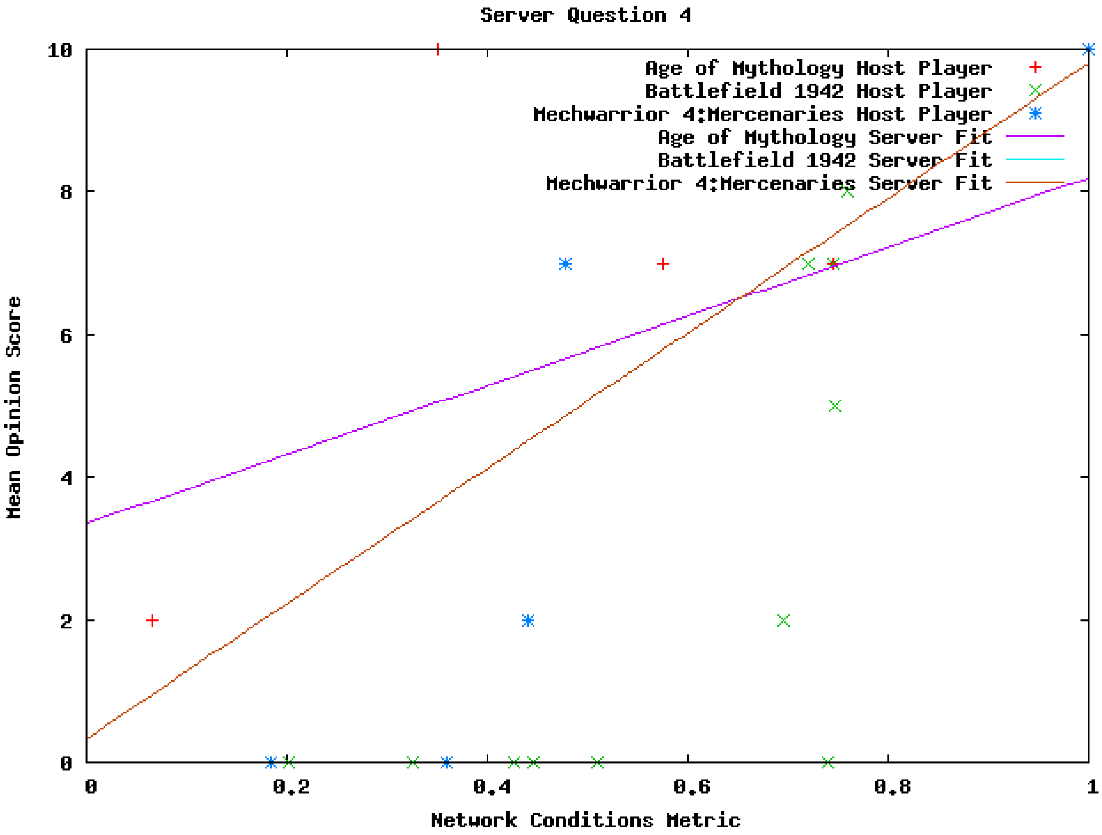


Figure 8.29: Server Question 4

Figure 8.29 shows the estimated response curves for Question 4 in the all 3 games for the server.

Server Question 5

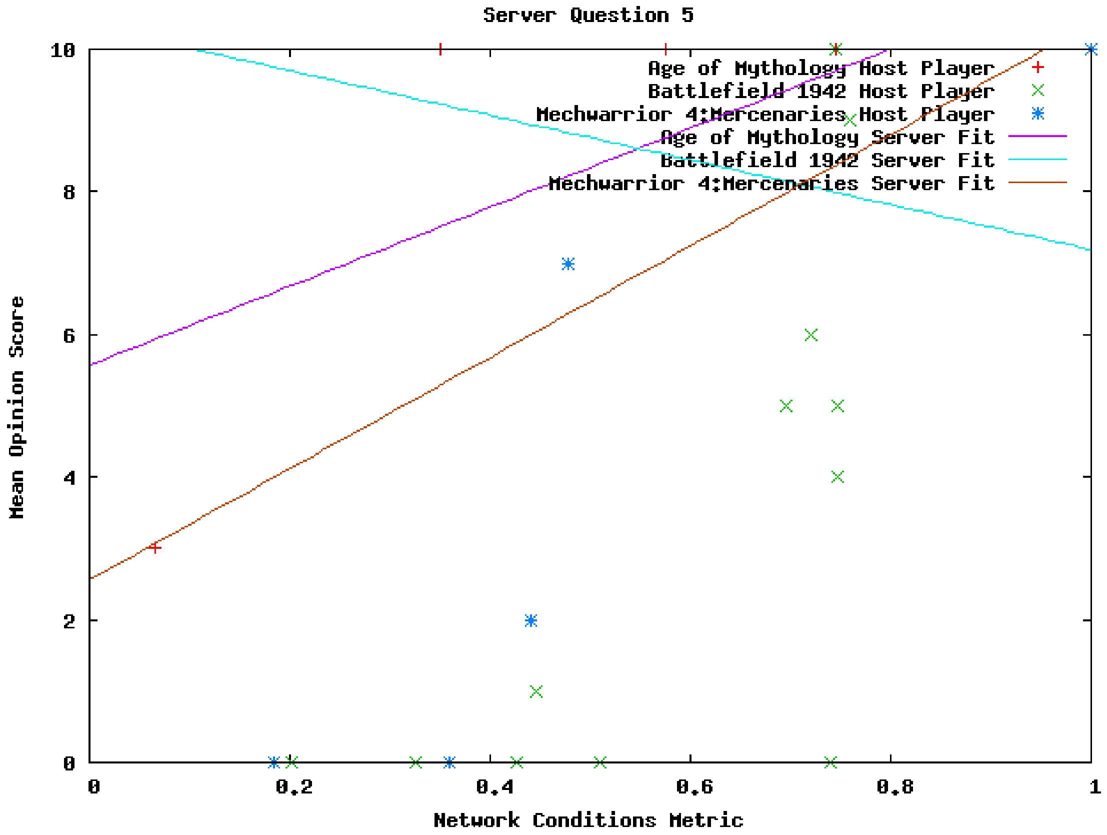


Figure 8.30: Server Question 5

Figure 8.30 shows the estimated response curves for Question 5 in the all 3 games for the server.

Server Question 6

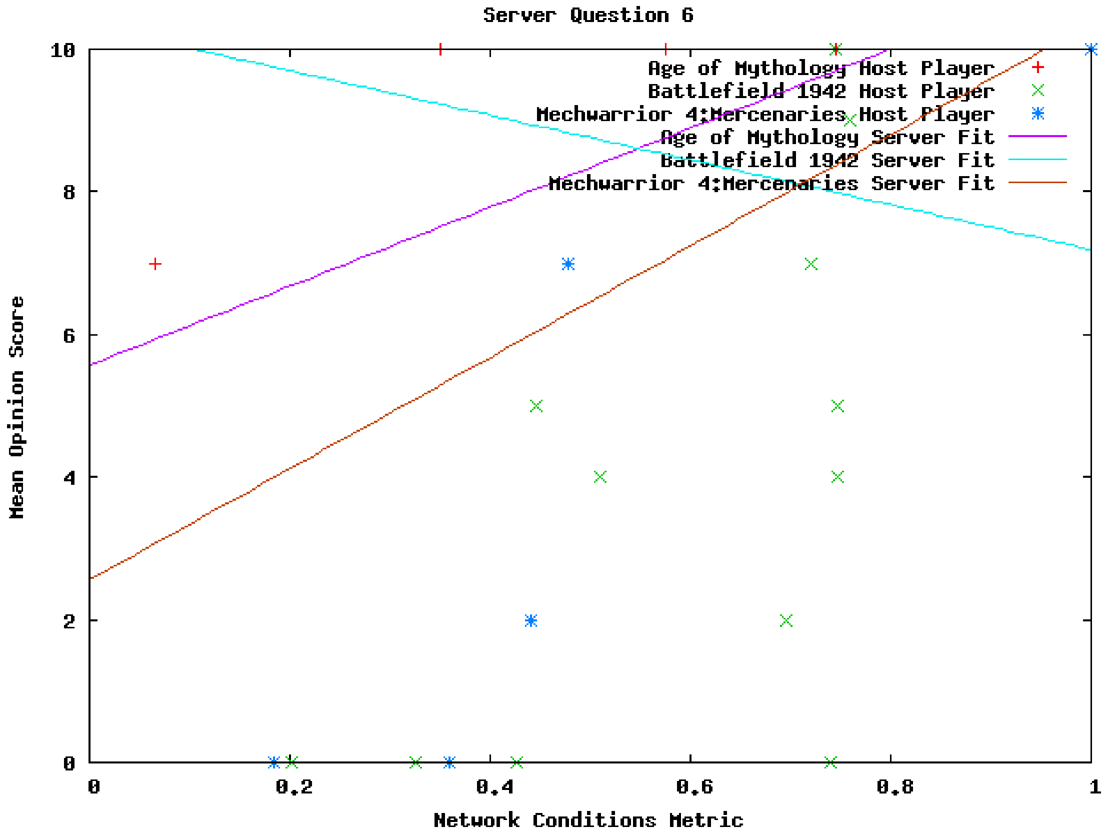


Figure 8.31: Server Question 6

Figure 8.31 shows the estimated response curves for Question 6 in the all 3 games for the server.

Server Question 7

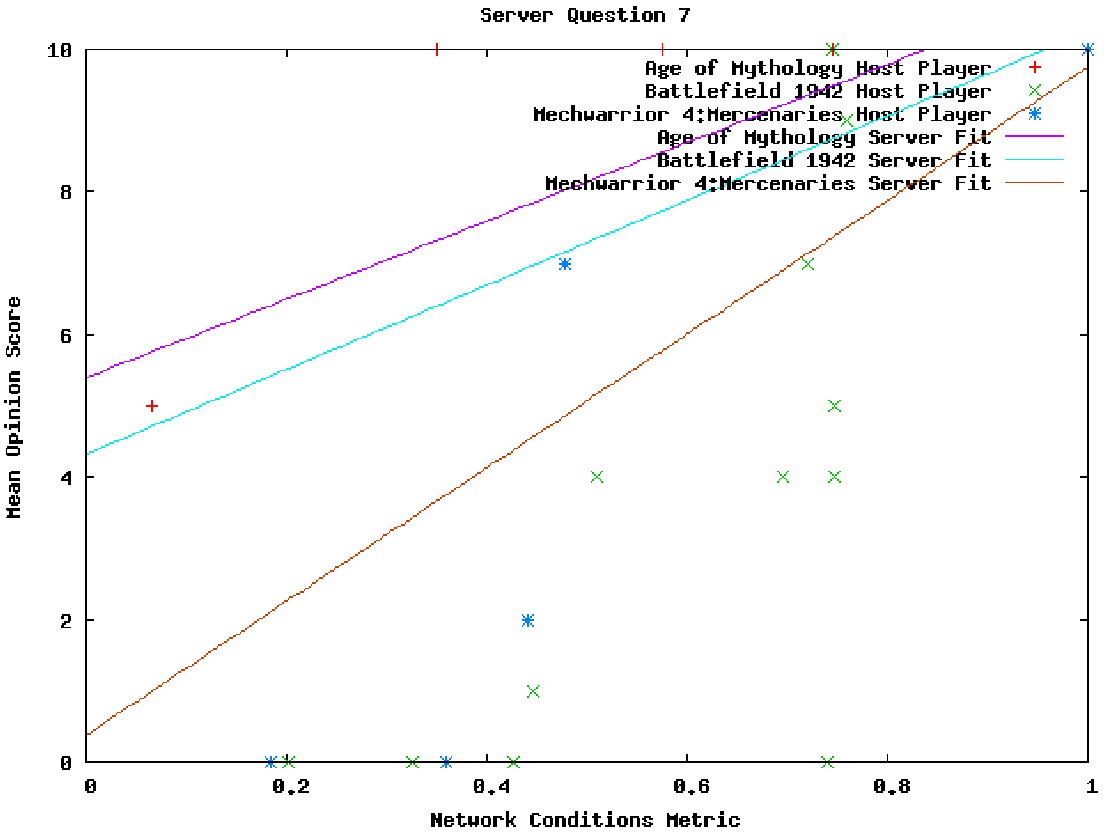


Figure 8.32: Server Question 7

Figure 8.32 shows the estimated response curves for Question 7 in the all 3 games for the server.

Overall Question 2

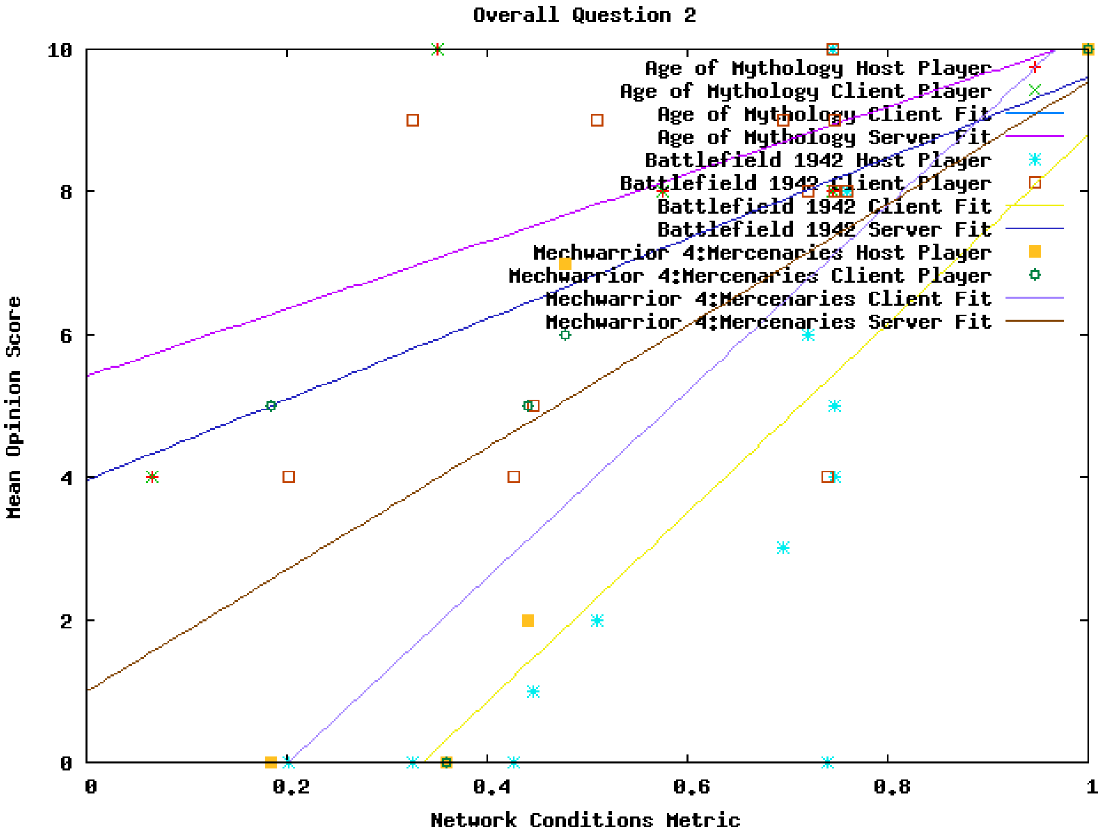


Figure 8.33: Overall Question 2

Figure 8.33 shows the estimated response curves for Question 2 in the all 3 games for the client and the server.

Overall Question 3

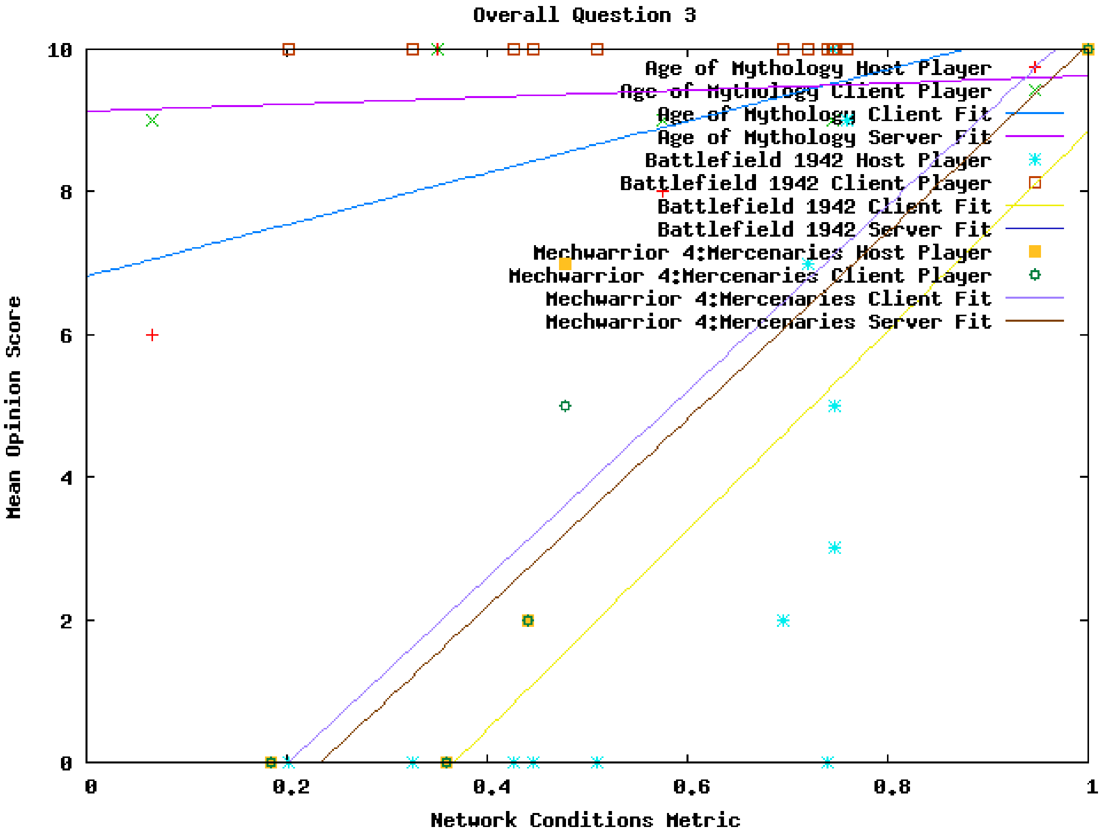


Figure 8.34: Overall Question 3

Figure 8.34 shows the estimated response curves for Question 3 in the all 3 games for the client and the server.

Overall Question 4

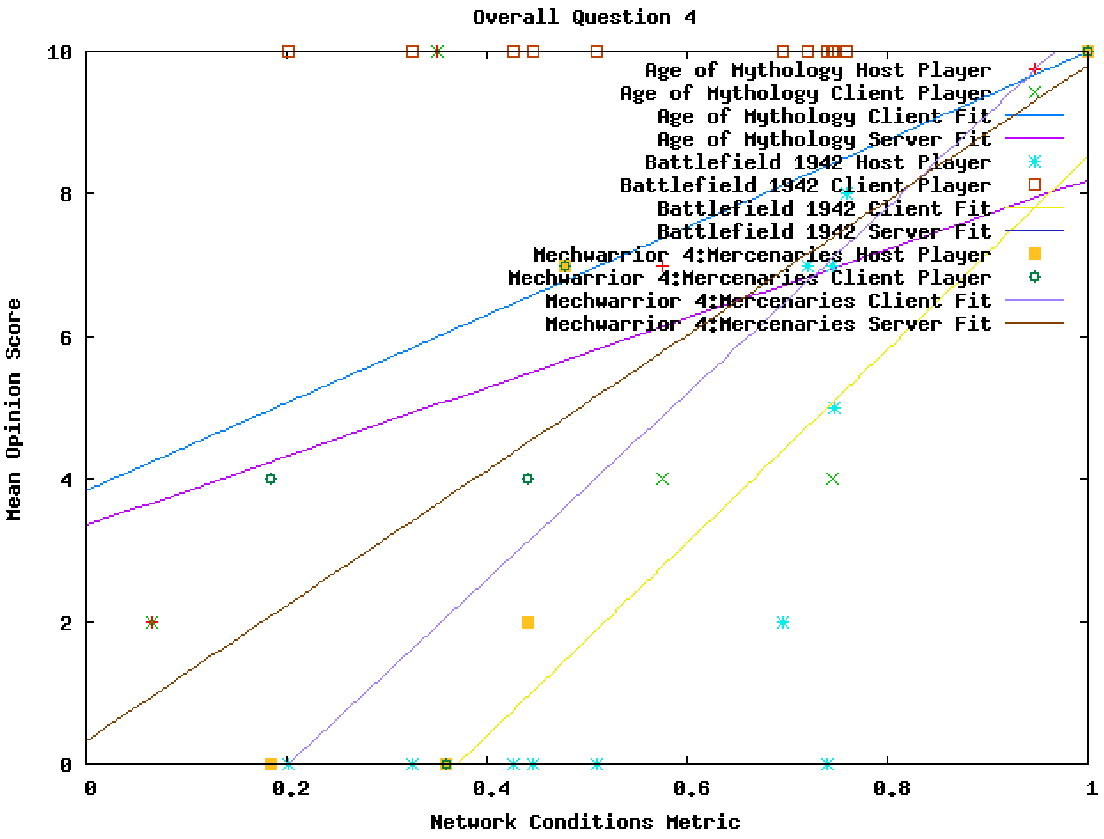


Figure 8.35: Overall Question 4

Figure 8.35 shows the estimated response curves for Question 4 in the all 3 games for the client and the server.

Overall Question 5

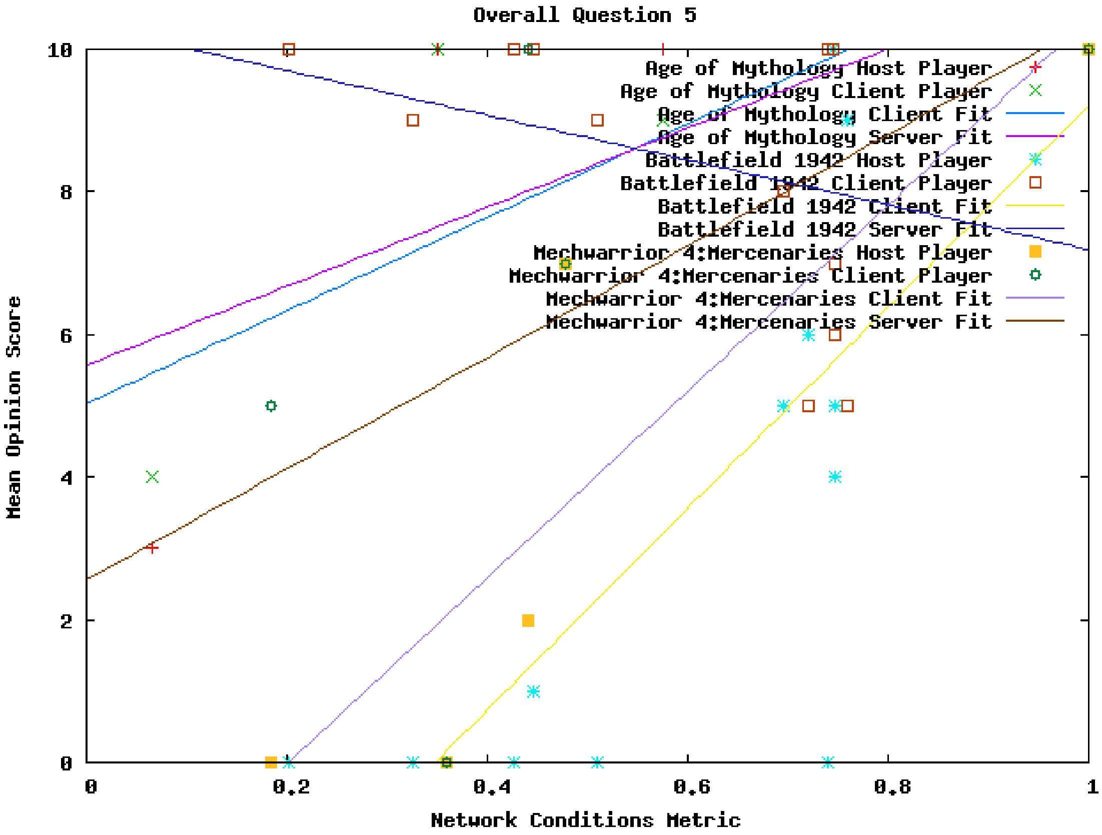


Figure 8.36: Overall Question 5

Figure 8.36 shows the estimated response curves for Question 5 in the all 3 games for the client and the server.

Overall Question 6

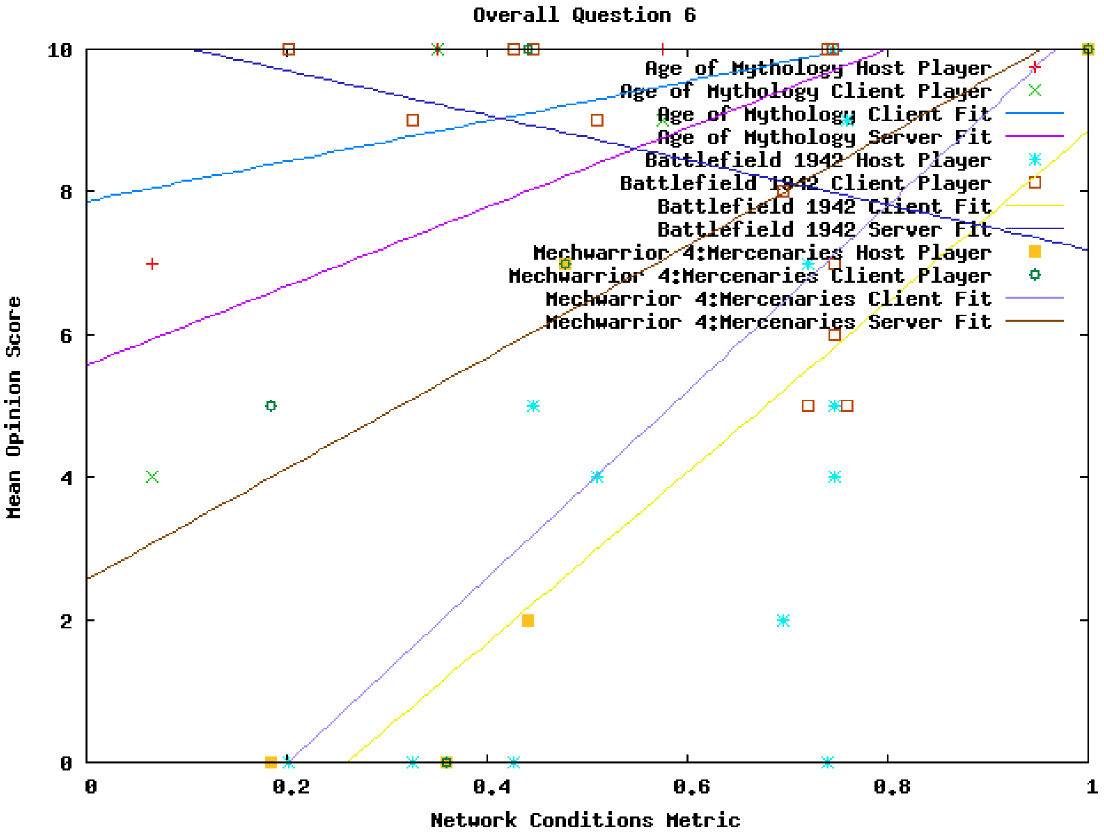


Figure 8.37: Overall Question 6

Figure 8.37 shows the estimated response curves for Question 6 in the all 3 games for the client and the server.

Overall Question 7

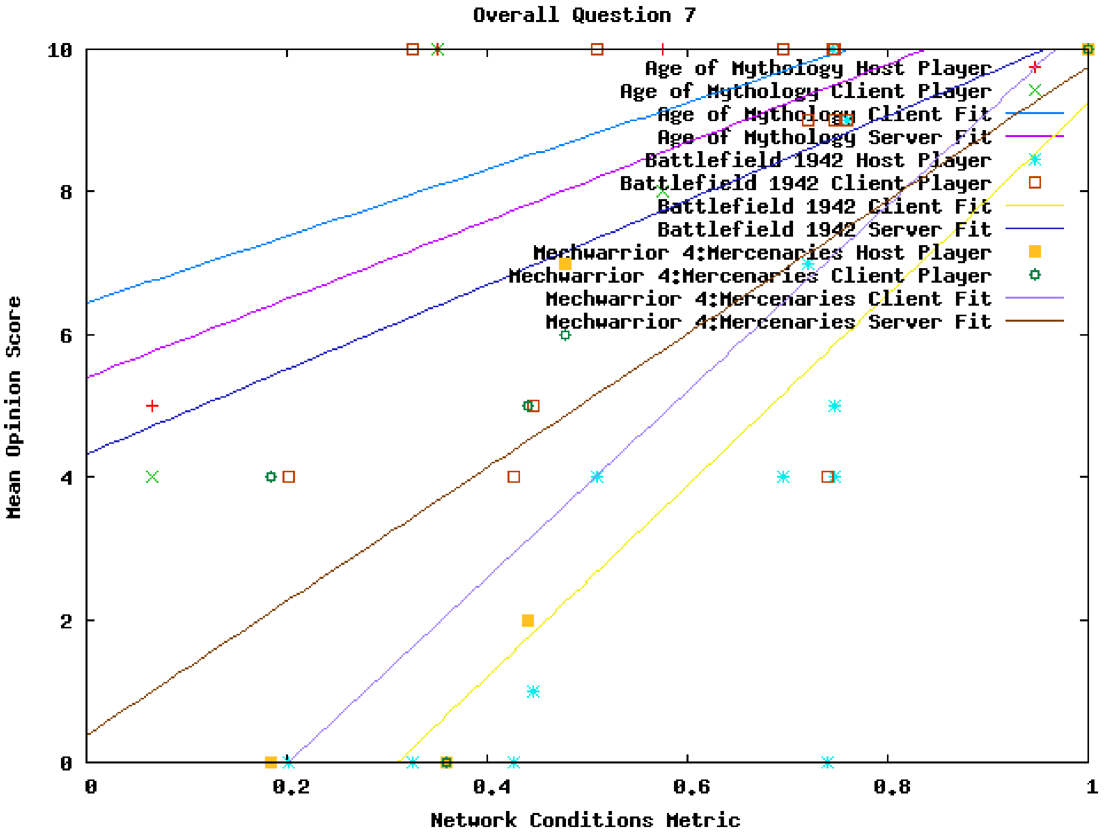


Figure 8.38: Overall Question 7

Figure 8.38 shows the estimated response curves for Question 7 in the all 3 games for the client and the server.

8.9 Analysis

8.9.1 Confirmation of Hypotheses

Hypothesis 1

Network Conditions have a direct impact on Game Player satisfaction

By regressing player response against technical quality of the communication conditions, we see that Hypothesis 1 is clearly confirmed. The t-Stat values for the coefficients are predominantly greater than 2.0 or less then -2.0 as shown in the tables 8.7, 8.8, 8.9, 8.10, 8.11 and 8.12. The results that are significant are underlined. Having a significant result, we conclude that there is a strong relationship between the player satisfaction and the network conditions.

In the game Age of Mythology, we have significant values across the 6 questions in the client intercept coefficient with the exception of Question 4. (Table 8.7) While on server end (Table 8.8), we also have the same 6 questions on the intercept as the client but we also have significant results on Questions 5, 6 and 7. It should be noted with reference to Table 8.2 that both players gave identical answers in Question 2.

Moving on to the game Battlefield 1942, we have significant values for the client (Table 8.9) on the intercept for Questions 2, 3, 4 and 5 with all questions for the the slope. On the Server end (Table 8.10), we have significant results on the intercept for Questions 2, 5, 6 and 7 and with Question 2 for the slope. Questions 3 and 4 have exactly the same answers given for all the questions and thus there is no error. (Table Table 8.4) This would be due to the small sample size.

Finally with the game Mechwarrior 4:Mercenaries, we have significant results for the client on the Slope (Tables 8.11). The server has significant results for Question 3 on the Intercept and Questions 3, 4 and 7 on the slope. 8.12

Hypothesis 2

Response curves are different for different Games.

We applied Student's t-Test to the difference between slope coefficients in the response curves to determine if these response curves are significantly different. Hypothesis 2 is clearly confirmed by the t-Distribution values underlined in the tables 8.13, 8.14, 8.15, 8.16, 8.17, 8.18, 8.19, 8.20 and 8.21.

In Table 8.14, the evidence of the difference between the client and the server is highly significant. During the course of the experiment, the client has recorded 3 connection breaks with the server while the server continued the game. The client was unable to play at all.

Comparing the different clients, the differences between Age of Mythology and Battlefield 1942 (Tables 8.16) are highly consistently significant at the 0.05 level. This is due to the different game models (genres). In Age of Mythology as with most Real

Time Strategy games, all computers are synchronized in the game world representation. On the other hand, Battlefield 1942 typical First Person Shooter's model, each client would try to flood the server with updates as much as possible and conduct a separate representation of the game world in each client. To a lesser extent, we see the same pattern of results in the Mechwarrior:4 Mercenaries/Age of Mythology comparisons in Table 8.18 and no significant differences when comparing Battlefield 1942 and Age of Mythology. (Table 8.17)

Comparing the different Servers in Tables 8.19, 8.20 and 8.21, we have some results that are significant at the 0.05 level. This is because all the different servers have a different communication model and/or protocol with their clients from each other.

Hypothesis 3

> *Different Genres have different Network Requirements*

Hypothesis 3 can be confirmed by the traffic patterns in the Appendix D. This is because the general shapes from the different sessions are similar for the same game. And these shapes are different to those from the other games. For example, the game Age of Mythology uses almost exclusively UDP Packets while the Game Mechwarrior:4 Mercenaries uses both TCP and UDP Packets. The game Battlefield 1942 has very erratic Data Transfer while Age of Mythology is very stable in terms of throughput.

Hypothesis 4

> *Because Game Traffic varies between games and network conditions, protocol stratification is useful.*

Hypothesis 4 cannot be proved at this time. However, the evidence from Hypothesis 2 and Hypothesis 3 supports this Hypothesis as highly plausible.

In the experiment and analysis, we have establish that player satisfaction is related to network performance. This is proved in the discussion in Hypothesis 2 by applying statistical analysis.

We can see the 2 similar games Mechwarrior 4:Mercenaries and Battlefield 1942 use very different network communication protocols, this is an indication that improvements to the protocol is plausible. This leads on to the idea that stratification can be applied to improve player satisfaction.

8.9.2 General Observations

The first observation is that if the client player disconnects from the server due to bad network performance, the affected player has poor satisfaction. This has occurred 3 times during the experiment with Battlefield 1942. This did not occur in other games. As a First Person Shooter, this is expected and is supported by current literature. [QML$^+$04] It can be noted as well to discover what parameters to adjust in the experiment. Packet Drops are discovered to be the most significant metric. The other point of note is that there is a statistical anomaly because the answers given by both players are exactly alike. We also can conclude that being on the Server or close to the server allows for high player satisfaction compared to the Real Time Strategy Game. This is in line with existing literature that suggest in a Real Time Strategy Game, the simulation step is synchronized across all computers. Because of the basic event loop in a Real Time Strategy Game, if one party has communications problems or slower computer hardware, all parities are almost equally affected. In a First Person Shooter, the game is simulated only on the server. [BT01]

8.10 Conclusions

The experiment was a pilot study in which the model is explored can be deemed a success generating lots of raw data for analysis. The importance of doing this exploratory study is to find out what is important before the doing a full study, Items like Network Conditions Settings needed to be tested to see if the settings are working. The parameters of the estimated models are review to be highly significant and there are clear and systematic differences between the games and genres explored. However one surprising point is that using settings similar to that of a GSM Network still allows satisfactory game play.

Full Study Details

The next step is to conduct a full study in which the sample size is increased to reduce errors. We expected the sample size to be about 5 to 10 times the current experiment. The player on the peer server sometimes gains an unfair advantage due to the game software design. This can be corrected if a symmetrical result is desired using a dedicated server if the game software allows. In addition, tools used to analyze the result should be automated as there are many repetitive calculations done to get the results.

Chapter 9

Conclusion

9.1 Remarks

The Mobile Gaming Sector of the Computer Industry is of steadily increasing importance. The modern consumer wants entertainment and fun on the go. However, there is no uniform single reference platform for which development can be targeted.

We have found that in multiplayer games, player satisfaction is highly sensitive to communication quality. Furthermore, different Gaming Genres have different requirements for communication quality. However, we have also observed that highly similar genres of game use different approaches to communication over networks and tackle communication problems quite differently, which suggests that the best approach for ensuring player satisfaction has not been identified yet.

Use of a Stratified Gaming Protocol potentially makes more efficient use of a poor communication channel. This is especially true in cases that are quite common in the mobile communication as current technology cannot provide the same Quality of Service as that wired links. Good protocol design has the potential to make a profound difference to the utility of mobile devices.

9.2 Further Work

The pilot study into the effects of poor communications quality on game playing satisfaction reported in this dissertation should be followed by a full study in order to consolidate the existing conclusions and further explore the relationship between technical quality and player satisfaction. This would be done using more games, players and sessions to gain accurate data and the hypothesis that protocol stratification can improve gaming satisfaction in bad network conditions could be further explored.

Other work would include the use of a stratified protocol in a game implementation and comparison of this with another implementation of the same game without stratification to estimate the improvement that can be obtained. A framework for this work has been set out in the experiment conducted in this dissertation.

Bibliography

[Ale03] Thor Alexander, editor. *Massively Multiplayer Game Development*. Charles River Media, 1st edition, 2003.

[Ast98] Astinus. A history of role-playing. `http://ptgptb.org/0001/history1.html`, 1998.

[BCL$^+$04] Tom Beigbeder, Rory Coughlan, Corey Lusher, John Plunkett, Emmanuel Agu, and Mark Claypool. The effects of loss and latency on user performance in unreal tournament 2003&#174;. In *SIGCOMM 2004 Workshops: Proceedings of ACM SIGCOMM 2004 workshops on NetGames '04*, pages 144–151, New York, NY, USA, 2004. ACM Press.

[Bik03] Jacco Bikker. Mobilecore: A cross-platform framework for arm-based mobile games. `http://www.gamasutra.com/features/20031117/bikker_01.shtml`, 2003.

[BT01] Paul Bettner and Mark Terrano. 1500 archers on a 28.8: Network programming in age of empires and beyond. `http://www.gamasutra.com/-features/20010322/terrano_02.htm`, 2001.

[Cis03] Cisco. *Internetworking Technologies Handbook*. Cisco Press, 4th edition, 2003.

[CS03] Mark Carson and Darrin Santay. Nist net: a linux-based network emulation tool. *SIGCOMM Comput. Commun. Rev.*, 33(3):111–126, 2003.

[Fox01] David Fox. Creating games using j2me. `http://www.gamasutra.com/resource_guide/20010917/fox_01.htm`, 2001.

[GZBS02] Jim Greer and Mine Zachary Booth Simpson. *Minimizing Latency in Real-time Strategy Games, Game Progarmming GEMS 3*, chapter 5.1, pages 488–495. Charles River Media, 1st edition, 2002.

[Ise02] Pete Isensee. *Secure Sockets, Game Progarmming GEMS 3*, chapter 5.6, pages 546–556. Charles River Media, 1st edition, 2002.

[Kle04] Leonard Kleinrock. The history of the internet. http://www.cs.ucla.edu/~lk/accomplishments.html, 2004.

[Kos02] Raph Koster. Online world timeline. http://www.legendmud.org/raph/gaming/mudtimeline.html, 2002.

[Mar01] Maury Markowitz. Spacewar - the first computer video game. really http://www3.sympatico.ca/maury/games/space/spacewar.html, 2001.

[Mic00] Sun Microsystems. *JSR-000037 Mobile Information Device Profile (MIDP)*. Sun Microsystems, 2000.

[Mic02] Sun Microsystems. *JSR-000118 Mobile Information Device Profile 2.0*. Sun Microsystems, 2002.

[MP03] Jessica Mulligan and Bridgette Patrovsky. *Developing Online Games: An Insider's Guide*. New Riders Games. New Riders, 1st edition, 2003.

[Myk00] Robert Mykland. *Palm OS Programming from the Ground Up*. Osborne, 1st edition, 2000.

[Pal03] Tommy Palm. The birth of the moblie mmog. http://www.gamasutra.com/resource_guide/20030916/palm_01.shtml, 2003.

[Pal04] PalmSource. *Palm OS Programmer's Companion*, volume 2. PalmSource, 1 edition, 2004.

[QML+04] Peter Quax, Patrick Monsieurs, Wim Lamotte, Danny De Vleeschauwer, and Natalie Degrande. Objective and subjective evaluation of the influence of small amounts of delay and jitter on a recent first person shooter game. In *SIGCOMM 2004 Workshops: Proceedings of ACM SIGCOMM 2004 workshops on NetGames '04*, pages 152–156, New York, NY, USA, 2004. ACM Press.

[SC04] Daniel Sanchez-Crespo. Handheld gaming marches forward at the gdc. http://www.gamasutra.com/gdc2004/features/20040326/postcard-sanchez_mobile.htm, 2004.

[Sep01] Lasse SeppŁnen. Designing mobile games for wap. http://www.gamasutra.com/resource_guide/20010917/seppanen_02.htm, 2001.

[SERZ02] Christian Schaefer, Thomas Enderes, Hartmut Ritter, and Marina Zitterbart. Subjective quality assessment for multiplayer real-time games. In *NETGAMES '02: Proceedings of the 1st workshop on Network and system support for games*, pages 74–78. ACM Press, 2002.

[SGB$^+$03] Nathan Sheldon, Eric Girard, Seth Borg, Mark Claypool, and Emmanuel Agu. The effect of latency on user performance in warcraft iii. In *NETGAMES '03: Proceedings of the 2nd workshop on Network and system support for games*, pages 3–14. ACM Press, 2003.

[Ste94] W. Richard Stevens. *TCP/IP ILLUSTRATED, VOLUME 1, THE PROTOCOLS*, volume 1. ADDISON-WESLEY PUBLISHING COMPANY, 1 edition, 1994.

[Way03] Peter Wayner. Policing online games: Digital currency. `http://www.gamasutra.com/features/20031010/wayner_01.shtml`, 2003.

Appendix A

Background

Beginning with the first computer game Spacewars! [Mar01] back in 1961 for the Digital Equipment Corperation PDP-1 (See Figure A.1), computer games have grown in size and complexitilty since then. In this Chapter, it is hoped that with a introduction to the genres of games.

A.1 Genres of Computer Games

As with the printed fiction or the motion picture, there are different genres. The computer game industry has many different genres of games and many are becoming cross genre, encompassing more then on traditional genre. This section helps to aid the reader in the more popular genres. An understanding of the different types of games would set the framework in which discussion and analysis of the networking backbone be in context.

A.1.1 Action Games

Fast movement and quick reflexes are two terms that can define the Action Games Genre. As one of the biggest and most exciting genre, with fast pace action, players often get caught up and lose all track of time. With many games styles falling under this genre, some more popular ones are highlighted here.

A very common style of Action games is the famous First Person Shooters or FPS as they are commonly called. First Person perspective is the term used to describe a game in which the screen displays what a person would see, as their character in the game. (Figure A.2 shows this perspective) FPS are games where in a first person perspective players move around a virtual environment using a representation of a weapon to "kill" monster or other players. Games in this genre can be said to have evolved from the humble beginnings of I.D. Soft Wolfenstein 3D (See Figure A.2) in 1992. This genre is driving the market with the latest offerings like Microsoft Ensemble Studios Halo, Sierra Online Half-Life's CounterStrike and Ubisoft's Far Cry

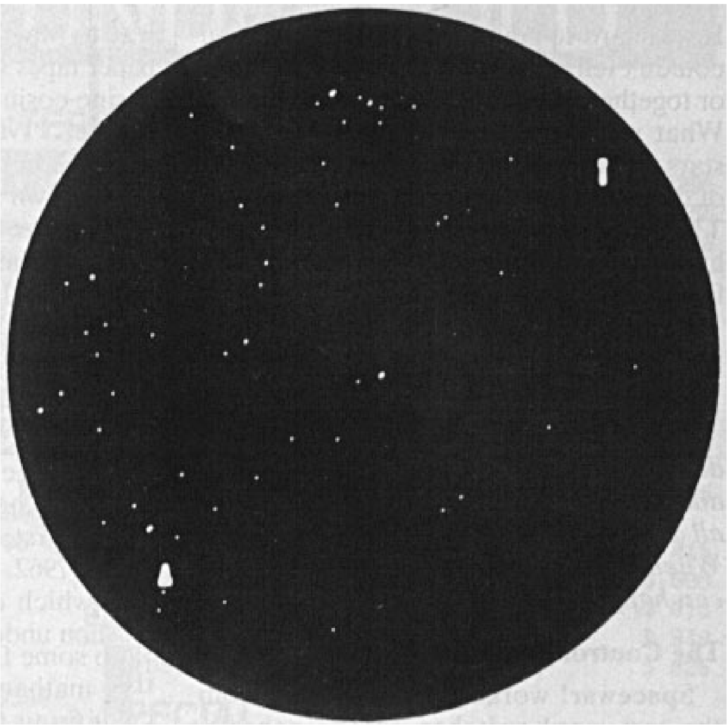


Figure A.1: Spacewar! on the PDP-1

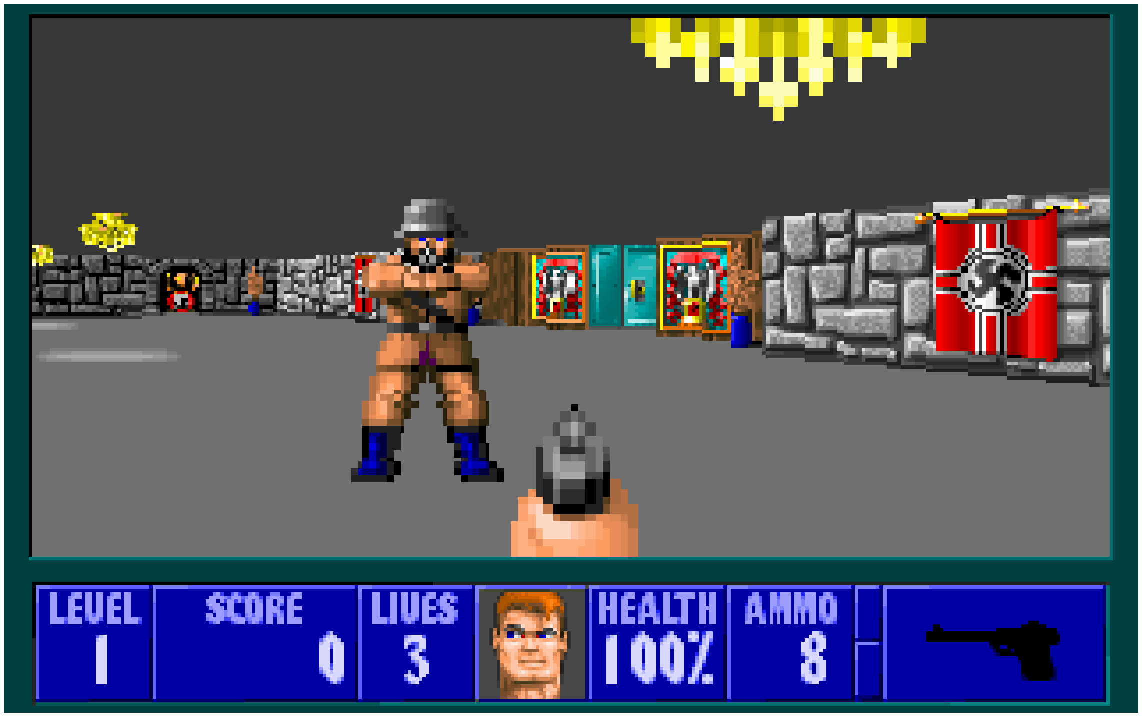


Figure A.2: Screenshot from Wolfenstein 3D

Figure A.3: Screenshot from Far Cry

(which currently top of line hardware is still considered lacking in full detailed mode. See Figure A.3) This style of game is increasingly driving players to make upgrades to their computer hardware to improve visual quality while playing.

There are many other types of action games aside from FPS. Another popular style is that of the third person fighting game. These are games where the player controlled character is represented. This style has many sub genres like the scrolling pane of Super Mario Brothers or the 3D movie licensed tie-in games.

A.1.2 Adventure Games

In The Grand Tour, or the murder mystery, an interesting story provides the basic framework of the game. Players in this genre explore a world and advance the story line. Famous classics from this genre include the Sierra Online King Quest Series & Leisure Suit Larry Series (Look out for the next installment). Another interesting possibility for this genre is that of the Internet Interactive Fiction (Another name for Adventure Games is that of Interactive Fiction). Sites like `http://www.addventure.`

Figure A.4: Screenshot from King’s Quest : Mask of Eternity

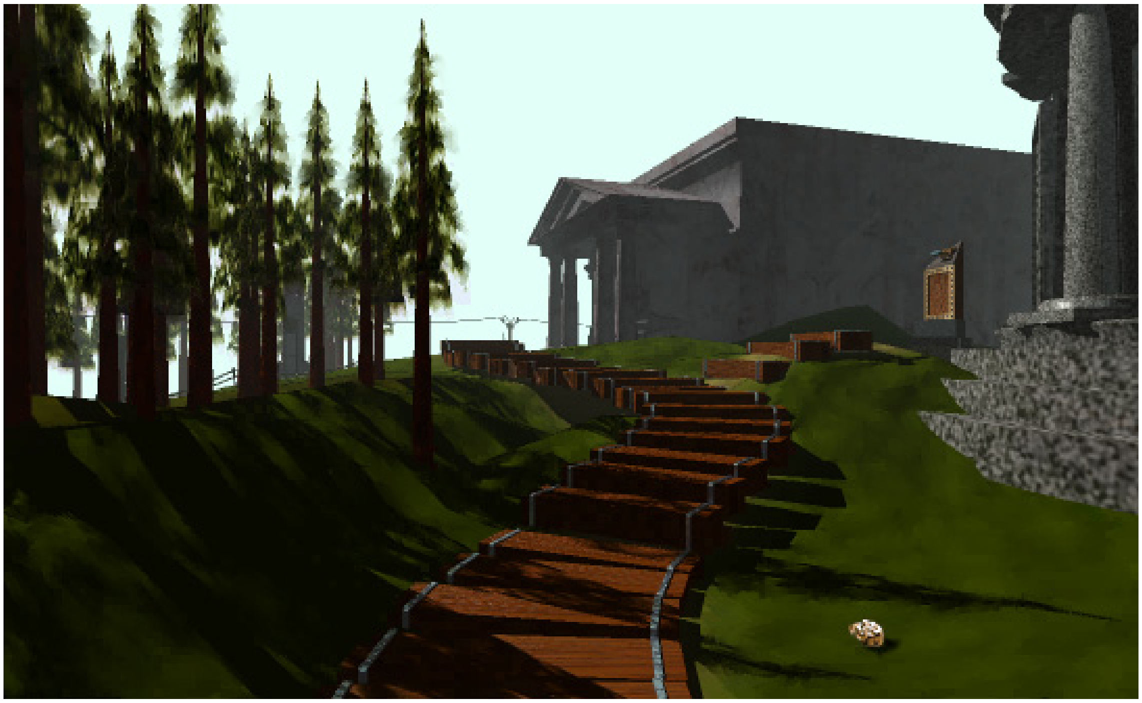


Figure A.5: Screenshot from Myst

com offer players user submitted linked stories.

A.1.3 Puzzle

The Genre of Puzzle types games is exemplified by Cyan World's Myst (See figure A.5) created in 1991. Gameplay in this Genre can be describe as the player solving a puzzle or series of puzzles. In the game Myst, the players explore a world while solving puzzles. As the player solves more puzzles, he or she advances the story-line, finally completing the set required number of puzzles in order to win the game.

This is not the only style of puzzle game, another very popular game in this genre is that of Tetris. (See Figure A.6) Puzzle Games are currently the most popular with mobile devices as they are simple and allow the player to have fun while on the go.

A.1.4 Role Playing Games

Back in 1970, Gary Gygax & Dave Arneson created a game which came to be known as Dungeon & Dragons. [Ast98] This is accepted as the first Pen & Paper Role Playing Game. The company that publishes Dungeon & Dragons is Tactical Studies Rules (Now part of the Wizards of The Coast Subsidiary of Hasbro). This genre can be mixed with the Adventure Game Genre. A key difference is that the character that the player controls (called the Player Character) retains experience & even weapons at the

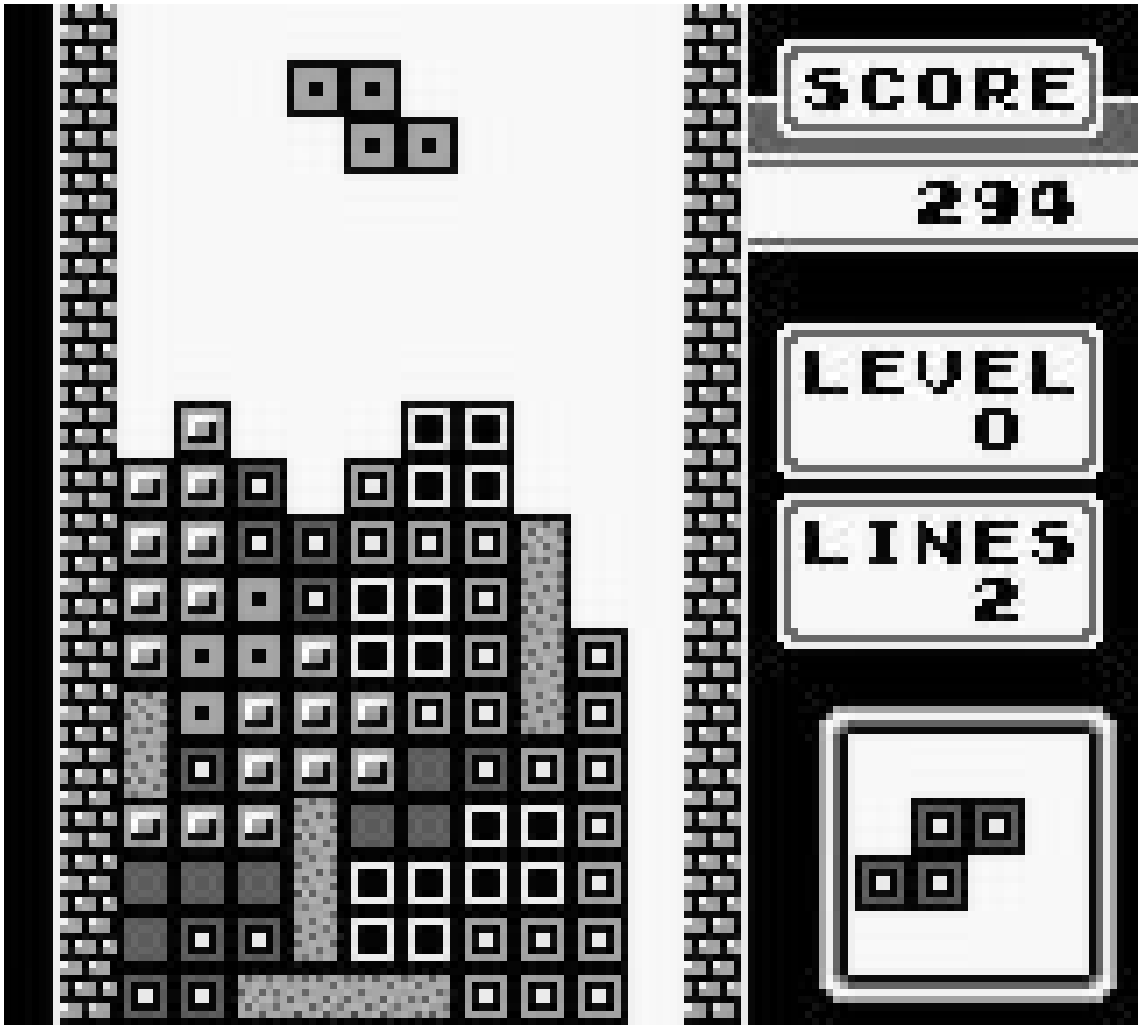


Figure A.6: Screenshot from Tetris

Figure A.7: Screenshot from Ultima Online

end of a session and this affects subsequent game playing sessions.

Early computer games in this genre are text based like Zork and Zork II. From that we get our graphical games like the Ultima series and evolving with the hardware capabilities of the Personal Computer. At about same time, we saw the introduction of MUD (Multi User Dungeons) which are text based multi user RPGs online. These games or environments typical require you to telnet into a server in which the server software provides an environment which your character can interact with and with other characters. The next logical step is that of the graphical MUD. An example of this is Furcaida, written as one of the original Graphical MUDs by some of the original developers of the Ultima series of games.

The development of the standalone RPG and its online version lead to a convergence of the 2 parallel evolution paths to give us our first MMORPG (Massive Multi-Player Online Role Playing Game), Ultima Online, by Origins. In these games, supporting 2 to 3 thousands concurrent users in a full graphical environment per server cluster, players interact with other human players as well as computer controlled characters and creatures.

These massive environments allow social interaction on a scale never seen before, for the common purposes of having fun. This is achieved by having computer con-

Figure A.8: Screenshot from Star Wars Racer

trolled creatures in which players can interact with each other as well as having tasks for players to complete. More then that, the software provides a virtual world in which interactions can be made on a human to human level. This rich environment is the attraction of such games. [Ale03]

A.1.5 Simulation

The field of simulation involves trying to model or recreate an environment for purposes of research, training or entertainment. Airlines spent millions of dollars to recreate the flight deck of their aircraft. Unable to do the real thing, be it flying a combat aircraft or be pod racer in the pod races described in the movie Star Wars I, immersed themselves in realistic or otherwise environments with their computer systems living out their fantasy. When discussing this genre, we must include a mention of Will Wright's creation of The Sims and Simcity. The Sims allows the player to control the lives of artificial characters in a user defined environment. Simcity, in its 4th incarnation now is a city building and management game.

Figure A.9: Screenshot from FIFA 2004 (Playstation)

A.1.6 Sports

The field of human sporting achievement has also been successfully translated into the computer arena. Sports Games encompass all normal sporting fields and some exciting uncommon sports like hunting and extreme sports. Such games translate the real world environment onto the computer where the player can play their sport of choice.

A.1.7 Strategy

War is a part of human history from the beginning. This genre of games are dominated by the 2 types of strategy game. The first is the turn based strategy. In this style of game, players enter their moves and end their turn (As in Chess and in its computer version). Then the next player, be it the Artificial Intelligence or another human player take shis or her turn.

The other type of game here is that of The Real Time Strategy. The RTS is a real-time game in which players battle it out in a virtual environment making moves and counter-moves. Famous RTS include Dune II for DOS and games like Blizzard Entertainment's Warcraft III. Often a term used with RTS is that of resource gathering.

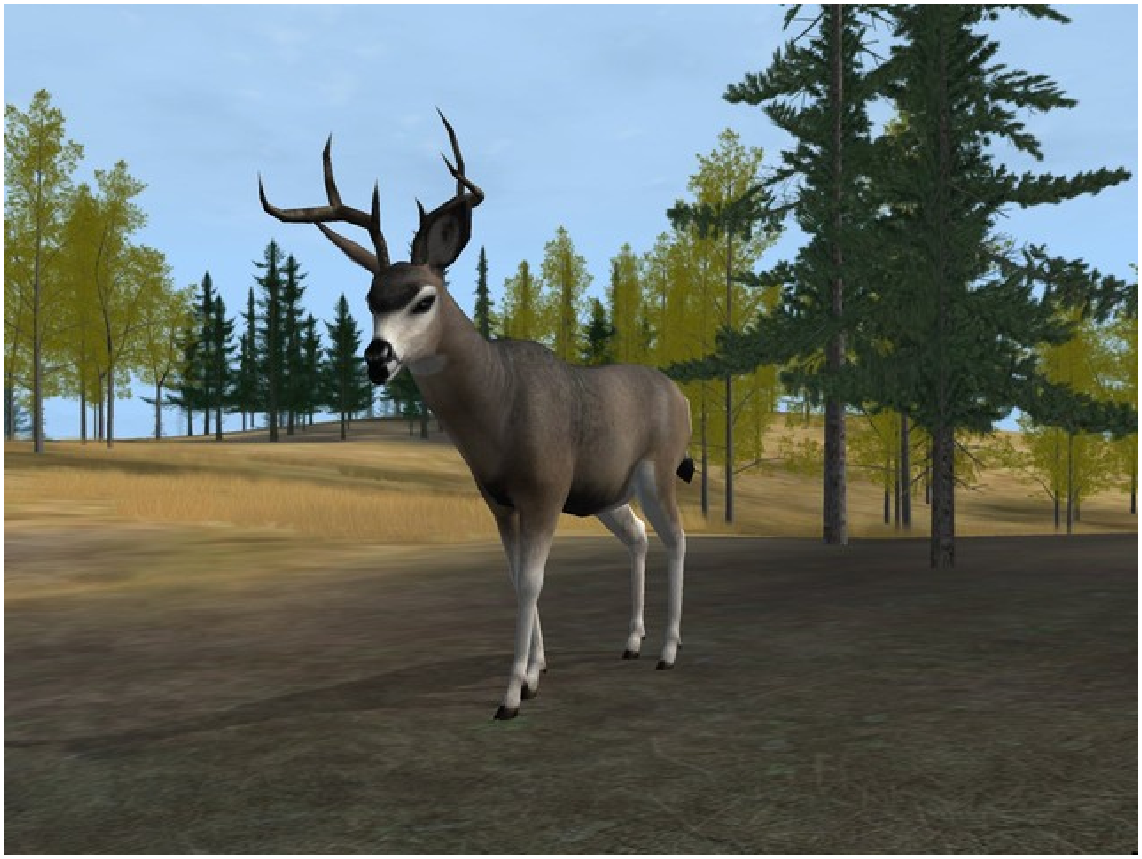
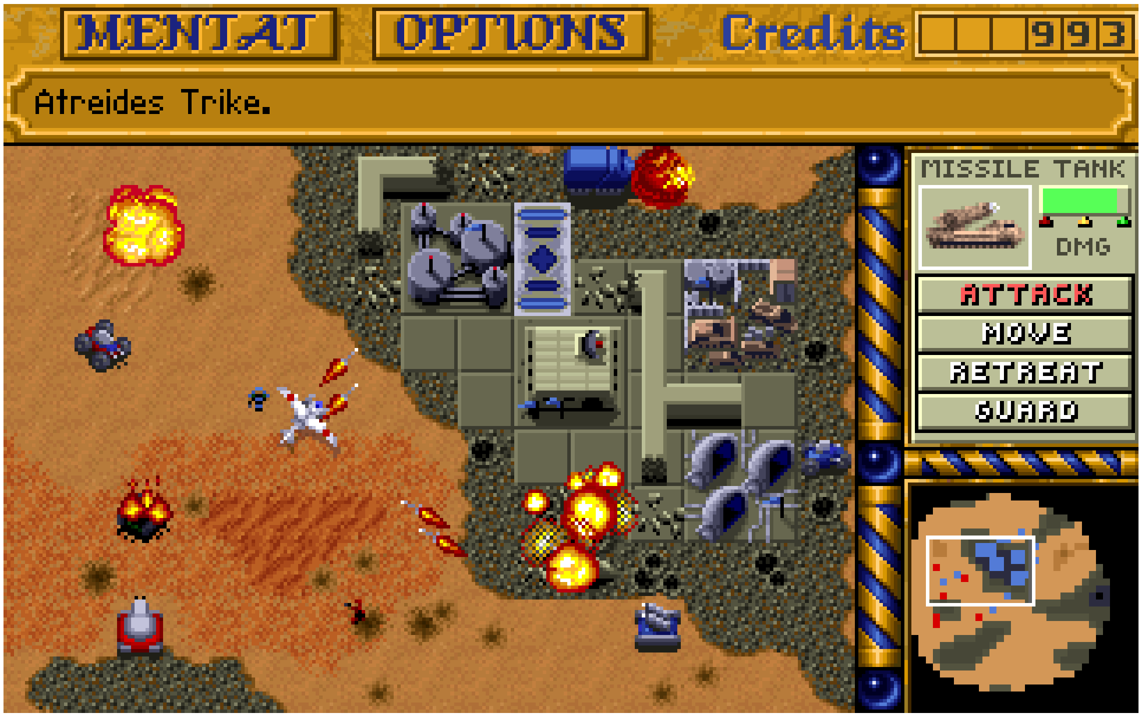


Figure A.10: Screenshot from Deer Hunter 2004



Atreides Trike.

Figure A.11: Screenshot from Dune 2

This term is the search and collection of raw materials in the game to build attacking units. This serves as a balance to that of the army building and attacking portion of the game.

Appendix B

Installation and Configuration of Nist.Net

This appendix is used to describe the installation process that was taken in the experiment discussed in earlier chapters. The documentation that Nist.Net has online is a bit vague and needs to be expanded to allow the reader to duplicate the environment. Valuable details gain through the process of trial and error are presented here.

B.1 Hardware Design

Nist.Net requires a Linux System running a 2.4 Kernel with 2 Network Cards. We used a Pentium 4 based with a 80gb Harddisk.

B.2 Operating System Installation

Gentoo Linux was selected as it offered a source compiled meta-distribution. This allows the selection of packages required and allows the tweaking of the system for high network performance.

B.2.1 Disk Partitioning

A 3 partition system was selected with the following.

- Boot Partition (100mb) Ext2
- Linux Swap Partition (1024 mb) Swap
- Root (Rest of Disk) Ext2

Ext2 was selected as it allows the easy recovery in case of failure, newer file systems are not supported in recovery systems.

B.2.2 Base Installation

A Stage 1 install was selected as it offers the most performance as very little is precompiled and everything is compile to GCC optimization settings given by us.

B.2.3 Kernel Installation

After trying many kernels and having compilation errors, the Pure Linux Source Kernel 2.4.30 was used. This maybe a localized issue. It must be repeated that Nist.Net only supports 2.0, 2.2 and 2.4 Kernels.

B.2.4 System Utilities Installation

As the Gentoo Linux installation process takes a long time, a small shell script as attached below to run without user input.

```
#!/bin/sh

cd /usr/portage
# scripts/bootstrap.sh -f
emerge system --fetchonly --emptytree

# scripts/bootstrap.sh

emerge ccache --fetchonly
emerge grub --fetchonly

emerge hotplug --fetchonly
emerge coldplug --fetchonly

emerge gentoo-sources --fetchonly
emerge genkernel --fetchonly

emerge vixie-cron --fetchonly
emerge sysklogd --fetchonly
emerge dhcpcd --fetchonly
emerge slocate --fetchonly

emerge ftp --fetchonly
emerge ntp --fethconly
emerge lynx --fetchonly

emerge gnome-light --fetchonly
emerge xscreensaver --fetchonly
emerge mozilla-firefox --fetchonly

emerge postfix --fetchonly
emerge mysql --fetchonly
emerge apache --fetchonly
emerge phpmyadmin --fetchonly

# emerge system --emptytree
emerge ccache

emerge gentoo-sources
emerge genkernel
emerge hotplug
emerge coldplug

genkernel all

emerge vixie-cron
emerge sysklogd
emerge dhcpcd
emerge slocate

emerge ftp
emerge ntp
emerge lynx

emerge gnome-light
emerge xscreensaver
emerge mozilla-firefox

emerge mrtg
emerge traffic-vis

emerge postfix
emerge mysql
emerge apache
emerge phpmyadmin

sh /root/after
```

B.2.5 X-Windows Configuration

The main point here is to add the Refresh rates for the Monitor in order to operate in higher resolutions.

```
Section "ServerLayout"
        Identifier     "X.org Configured"
        Screen      0  "Screen0" 0 0
        InputDevice    "Mouse0" "CorePointer"
        InputDevice    "Keyboard0" "CoreKeyboard"
EndSection

Section "Files"
        RgbPath      "/usr/lib/X11/rgb"
        ModulePath   "/usr/lib/modules"
        FontPath     "/usr/share/fonts/misc/"
        FontPath     "/usr/share/fonts/TTF/"
        FontPath     "/usr/share/fonts/Type1/"
        FontPath     "/usr/share/fonts/CID/"
        FontPath     "/usr/share/fonts/75dpi/"
        FontPath     "/usr/share/fonts/100dpi/"
EndSection

Section "Module"
        Load  "glx"
        Load  "dbe"
        Load  "dri"
        Load  "extmod"
        Load  "record"
        Load  "xtrap"
        Load  "freetype"
        Load  "type1"
EndSection

Section "InputDevice"
        Identifier  "Keyboard0"
        Driver      "kbd"
EndSection

Section "InputDevice"
        Identifier  "Mouse0"
        Driver      "mouse"
        Option            "Protocol" "auto"
        Option            "Device" "/dev/mouse"
EndSection

Section "Monitor"
        Identifier   "Monitor0"
        VendorName   "Monitor Vendor"
        ModelName    "Monitor Model"
        HorizSync        31.5 - 90.0
        VertRefresh        40 - 75
EndSection

Section "Device"
        ### Available Driver options are:-
        ### Values: <i>: integer, <f>: float, <bool>: "True"/"False",
        ### <string>: "String", <freq>: "<f> Hz/kHz/MHz"
        ### [arg]: arg optional
        #Option     "SWcursor"                   # [<bool>]
        #Option     "HWcursor"                   # [<bool>]
        #Option     "NoAccel"                    # [<bool>]
        #Option     "ShadowFB"                   # [<bool>]
        #Option     "UseFBDev"                   # [<bool>]
        #Option     "Rotate"                     # [<str>]
        #Option     "VideoKey"                   # <i>
        #Option     "FlatPanel"                  # [<bool>]
        #Option     "FPDither"                   # [<bool>]
        #Option     "CrtcNumber"                 # <i>
        #Option     "FPScale"                    # [<bool>]
        #Option     "FPTweak"                    # <i>
        Identifier  "Card0"
        Driver      "nv"
        VendorName  "nVidia Corporation"
        BoardName   "NV11 [GeForce2 MX/MX 400]"
        BusID       "PCI:1:0:0"
EndSection

Section "Screen"
        Identifier "Screen0"
        Device     "Card0"
        Monitor    "Monitor0"
        DefaultDepth 16
        SubSection "Display"
                Viewport   0 0
                Depth     8
                Modes "1024x768"
        EndSubSection
        SubSection "Display"
                Viewport   0 0
                Depth     15
                Modes "1024x768"
        EndSubSection
        SubSection "Display"
                Viewport   0 0
                Depth     16
                Modes "1024x768"
        EndSubSection
        SubSection "Display"
                Viewport   0 0
                Depth     24
                Modes "1024x768"
        EndSubSection
EndSection
```

B.3 Network Configuration

The /etc/conf.d/net file is used to setup the fixed IP addresses.

```
# /etc/conf.d/net:
# $Header: /home/cvsroot/gentoo-src/rc-scripts/etc/conf.d/net,v 1.7 2002/11/18 19:39:22 azar

# Global config file for net.* rc-scripts

# This is basically the ifconfig argument without the ifconfig $iface
#
iface_eth2="192.168.2.200 broadcast 192.168.2.255 netmask 255.255.255.0"
iface_eth1="192.168.1.1 broadcast 192.168.1.255 netmask 255.255.255.0"
iface_eth0="192.168.0.1 broadcast 192.168.0.255 netmask 255.255.255.0"

#iface_eth1="207.170.82.202 broadcast 207.0.255.255 netmask 255.255.0.0"

# For DHCP set iface_eth? to "dhcp"
# For passing options to dhcpcd use dhcpcd_eth?
#
#iface_eth0="dhcp"
#dhcpcd_eth0="..."

# For adding aliases to a interface
#
#alias_eth0="192.168.0.3 192.168.0.4"

# NB:  The next is only used for aliases.
#
# To add a custom netmask/broadcast address to created aliases,
# uncomment and change accordingly.  Leave commented to assign
# defaults for that interface.
#
```

```
#broadcast_eth0="192.168.0.255 192.168.0.255"
#netmask_eth0="255.255.255.0 255.255.255.0"


# For setting the default gateway
#
gateway="eth2/192.168.2.1"
```

The sysctl.conf file is used to enable packet forwarding which is required to turn the machine into a router.

```
# /etc/sysctl.conf:
# $Header: /home/cvsroot/gentoo-src/rc-scripts/etc/sysctl.conf,v 1.3 2002/11/18 19:39:22 azarah

# Disables packet forwarding
net.ipv4.ip_forward = 1
# Disables IP dynaddr
#net.ipv4.ip_dynaddr = 0
# Disable ECN
net.ipv4.tcp_ecn = 0
# Enables source route verification
#net.ipv4.conf.default.rp_filter = 1
# Disables the magic-sysrq key
#kernel.sysrq = 0

net.ipv4.conf.all.proxy_arp = 1
```

B.4 Nist.Net Installation

Nist.Net has very little information on how to compile and setup. Attach are the 2 files which needs to be modified in order to compile.

B.4.1 Configuration Details

The first would be the Config file which the Kernel Source Directories need to be set.

```
# $Header$

# This Makefile passes common flags to subdirectories.
# Mark Carson, NIST/UMCP
#  1/2000

ARCH = i386

.EXPORT_ALL_VARIABLES:

# 1. This should point to the top of the Linux kernel source tree:
TOPDIR = /usr/src/linux
HPATH = $(TOPDIR)/include

# 2. Device node names and major numbers -- edit here and recompile if needed
DEVHITBOX = /dev/hitbox
DEVNISTNET = /dev/nistnet
HITMAJOR = 62        # .0625 = 1/16, Lina Inverse is 16 in NEXT
HITMINOR = 0
NISTNETMAJOR = 62
NISTNETMINOR = 1
DEVMUNGEBOX = /dev/mungebox
DEVSPYBOX = /dev/spybox
MUNGEMAJOR = 63
SPYMAJOR = 64

# 3. Device defines and feature configs.

# a. By default, we include ECN (explicit congestion notification) setting
# support, but not COS (class of service) selection support.  Change them
# here if you wish.
ECN = -DCONFIG_ECN
COS = # -DCONFIG_COS

# b. How do you want to do bandwidth delays?  Bandwidth delays amount to
# pretending that a packet takes some period of time to send.  The
# question then is, when should the packet actually be sent during this
# interval - at the beginning (default) middle, or end?  Uncomment the
# desired choice.
BDELAY = -DCONFIG_DELAYSTART
#BDELAY = -DCONFIG_DELAYMIDDLE
#BDELAY = -DCONFIG_DELAYEND

# c. How "aggressive" do we want to be about using the RTC (real-time clock)?
# By defining CONFIG_RTC_AGGRESSIVE, we will seize control of the RTC IRQ
# regardless of whether somebody else (the /dev/rtc driver) has it.
# For a modular rtc (possible with 2.4 kernels) this is unnecessary,
# since you can just rmmod rtc before starting NIST Net.  But for
# 2.0 and 2.2 kernels, there’s really not much choice if the rtc has
# been compiled into the kernel (which is usually the case).
RTC_HANDLING = -DCONFIG_RTC_AGGRESSIVE

# d. Which delay distribtuion to use? By default, it’s experimental.
# With the new version, this will be expanded to cover other delay
# algorithms (e.g., MWM).

DISTRIBUTION= -DUSE_EXPERIMENTAL -DDISTRIBUTION_NAME="\"experimental\""
ALPHA= -DPARETOALPHA=3.2

# ----- The rest of this should not require modification under normal
# ----- circumstances!
# ---------------------------------------------------------------------

DEVDEFS= -DDEVHITBOX=\"$(DEVHITBOX)\" -DHITMAJOR=$(HITMAJOR) -DHITMINOR=$(HITMINOR) \
        -DDEVNISTNET=\"$(DEVNISTNET)\" -DNISTNETMAJOR=$(NISTNETMAJOR)  -DNISTNETMINOR=$(NISTNET
        -DDEVMUNGEBOX=\"$(DEVMUNGEBOX)\" -DMUNGEMAJOR=$(MUNGEMAJOR) \
        -DDEVSPYBOX=\"$(DEVSPYBOX)\" -DSPYMAJOR=$(SPYMAJOR) \
        $(ECN) $(COS) $(BDELAY) $(RTC_HANDLING)

# Universal flags:
CROSS_COMPILE   =
# Can do $(PWD)/include with tcsh but not ksh??
OURINCS = ../include

CC        = $(CROSS_COMPILE)gcc

COMMONCFLAGS        = -I. -I$(OURINCS) -Wall -Wstrict-prototypes -fno-strength-reduce -pipe -ma

AS        =$(CROSS_COMPILE)as
LD        =$(CROSS_COMPILE)ld
AR        =$(CROSS_COMPILE)ar
NM        =$(CROSS_COMPILE)nm
STRIP        =$(CROSS_COMPILE)strip
MAKE        =make
AWK        =gawk

# Kernel-level flags:
KERNCC        =$(CC) -D__KERNEL__
KERNCPP        =$(KERNCC) -E
KERNELDEBUGFLAGS = -O2 -fomit-frame-pointer
KERNCFLAGS        = -I$(HPATH) $(COMMONCFLAGS) $(KERNELDEBUGFLAGS) -DMODULE

MODFLAGS = -DMODULE
```

```
MAKING_MODULES=1

# User-level flags:
USERDEBUGFLAGS = -O
CFLAGS         = $(COMMONCFLAGS) $(USERDEBUGFLAGS)
```

Second would be the widget system used as shown.

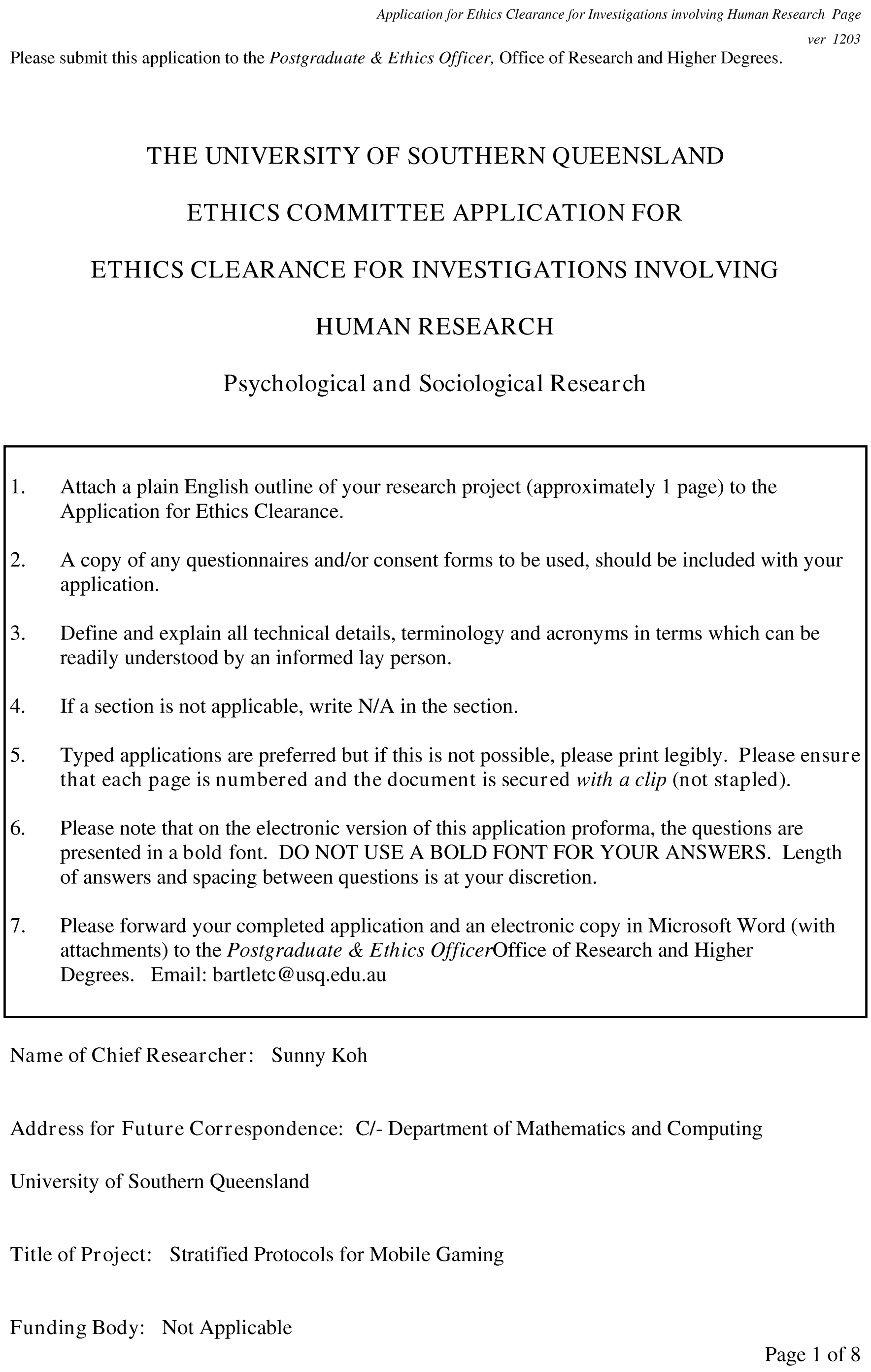
```
# $Header: /src/carson/nistnet/monitor/RCS/Imakefile,v 1.4 2000/03/22 16:08:44 carson Exp ca

EXTRA_DEFINES = $(DEVDEFS)

TEXTFIELD = TextField-1.01

FRAME = Frame-1.0

EXTRA_INCLUDES = -I$(TEXTFIELD) -I$(FRAME) -I$(INCROOT)/X11/Xaw -I$(INCROOT)/X11/Xmu -I.. -I

EXTRALIB = ../lib/libnistnet.a

EXTRAOBJS = $(TEXTFIELD)/TextField.o $(FRAME)/Frame.o $(FRAME)/Gcs.o

OBJS1 = hitmonitor.o xhitutil.o tabchain.o

SRCS1 = hitmonitor.c xhitutil.c tabchain.c

PROGRAMS = xnistnet


# You can use any of the Xaw-compatible widget sets for the user interface.
# Some possibilities:
# 1. -lneXtaw - "NeXt-like" Athena widgets (best scrollbars of the bunch)
# 2. -lXaw3d - default choice, ok appearance, but odd mouse behavior
# 3. -lXaw - original Xaw widget set, for the hardcore only
# Set your choice here:
OURXAWLIB = -lXaw

# Recursive stuff
#define PassCDebugFlags 'CDEBUGFLAGS=$(CDEBUGFLAGS)'

MakefileSubdirs($(TEXTFIELD) $(FRAME))

MakeSubdirs($(TEXTFIELD) $(FRAME))

CleanSubdirs($(TEXTFIELD) $(FRAME))
```

```
# Actual build target
OBJS = $(OBJS1) $(EXTRAOBJS)
LOCAL_LIBRARIES = $(OURXAWLIB) $(XTOOLLIB) $(XMULIB) $(XLIB) $(EXTRALIB) -lm

ComplexProgramTargetNoMan(xnistnet)


DependTarget()
```

With this, a basic installation of Nist.Net would be created.

Appendix C

USQ Ethics Clearance



Application for Ethics Clearance for Investigations involving Human Research Page

ver 1203

Please submit this application to the *Postgraduate & Ethics Officer,* Office of Research and Higher Degrees.

THE UNIVERSITY OF SOUTHERN QUEENSLAND

ETHICS COMMITTEE APPLICATION FOR

ETHICS CLEARANCE FOR INVESTIGATIONS INVOLVING

HUMAN RESEARCH

Psychological and Sociological Research

1. Attach a plain English outline of your research project (approximately 1 page) to the Application for Ethics Clearance.
2. A copy of any questionnaires and/or consent forms to be used, should be included with your application.
3. Define and explain all technical details, terminology and acronyms in terms which can be readily understood by an informed lay person.
4. If a section is not applicable, write N/A in the section.
5. Typed applications are preferred but if this is not possible, please print legibly. Please ensure that each page is numbered and the document is secured *with a clip* (not stapled).
6. Please note that on the electronic version of this application proforma, the questions are presented in a bold font. DO NOT USE A BOLD FONT FOR YOUR ANSWERS. Length of answers and spacing between questions is at your discretion.
7. Please forward your completed application and an electronic copy in Microsoft Word (with attachments) to the *Postgraduate & Ethics Officer*Office of Research and Higher Degrees. Email: bartletc@usq.edu.au

Name of Chief Researcher: Sunny Koh

Address for Future Correspondence: C/- Department of Mathematics and Computing

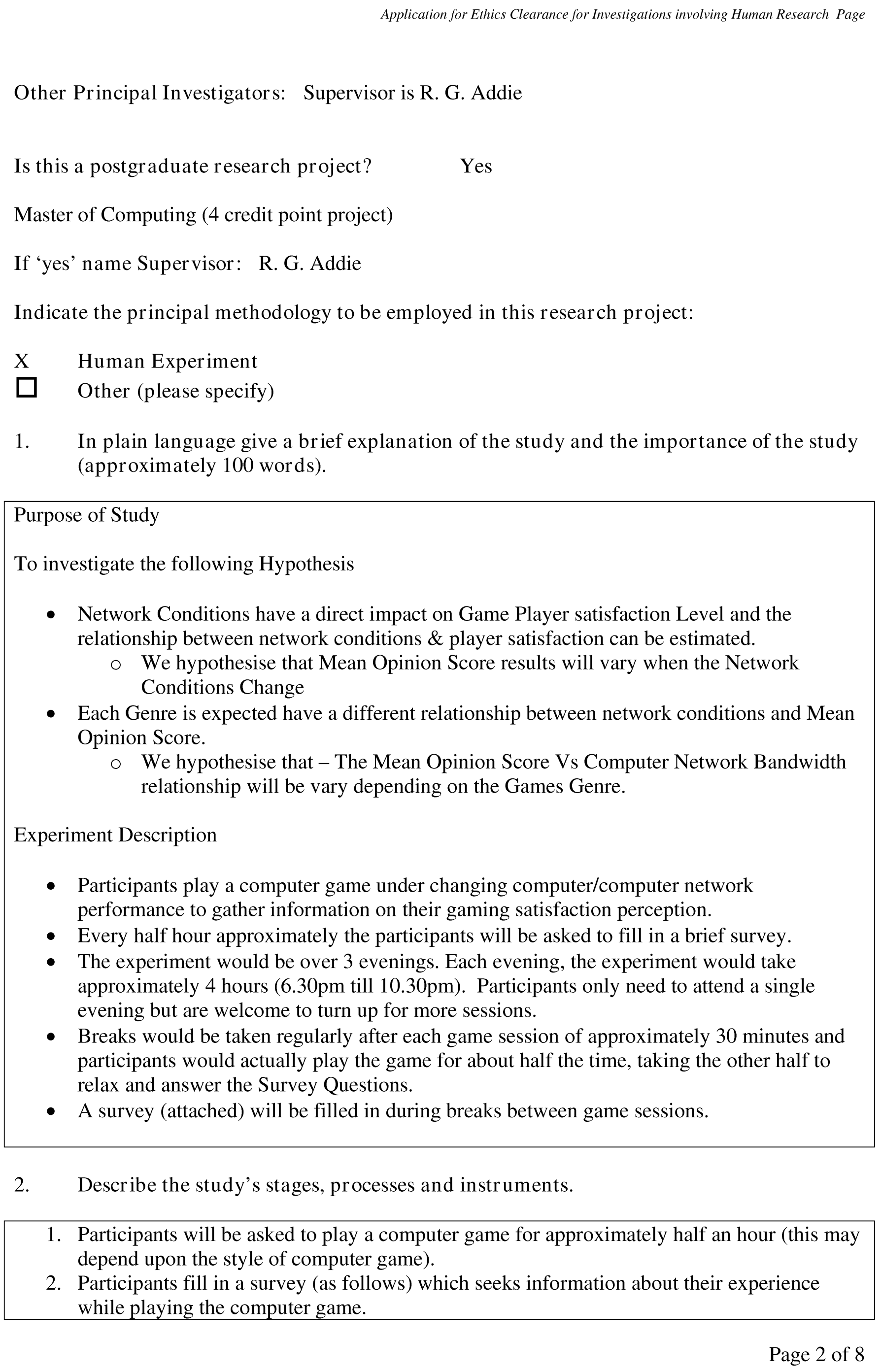
University of Southern Queensland

Title of Project: Stratified Protocols for Mobile Gaming

Funding Body: Not Applicable

Page 1 of 8

Figure C.1: USQ Ethics Clearance Page 1



Other Principal Investigators: Supervisor is R. G. Addie

Is this a postgraduate research project? Yes

Master of Computing (4 credit point project)

If 'yes' name Supervisor: R. G. Addie

Indicate the principal methodology to be employed in this research project:

X Human Experiment
□ Other (please specify)

1. In plain language give a brief explanation of the study and the importance of the study (approximately 100 words).

Purpose of Study

To investigate the following Hypothesis

- Network Conditions have a direct impact on Game Player satisfaction Level and the relationship between network conditions & player satisfaction can be estimated.
  - We hypothesise that Mean Opinion Score results will vary when the Network Conditions Change
- Each Genre is expected have a different relationship between network conditions and Mean Opinion Score.
  - We hypothesise that – The Mean Opinion Score Vs Computer Network Bandwidth relationship will be vary depending on the Games Genre.

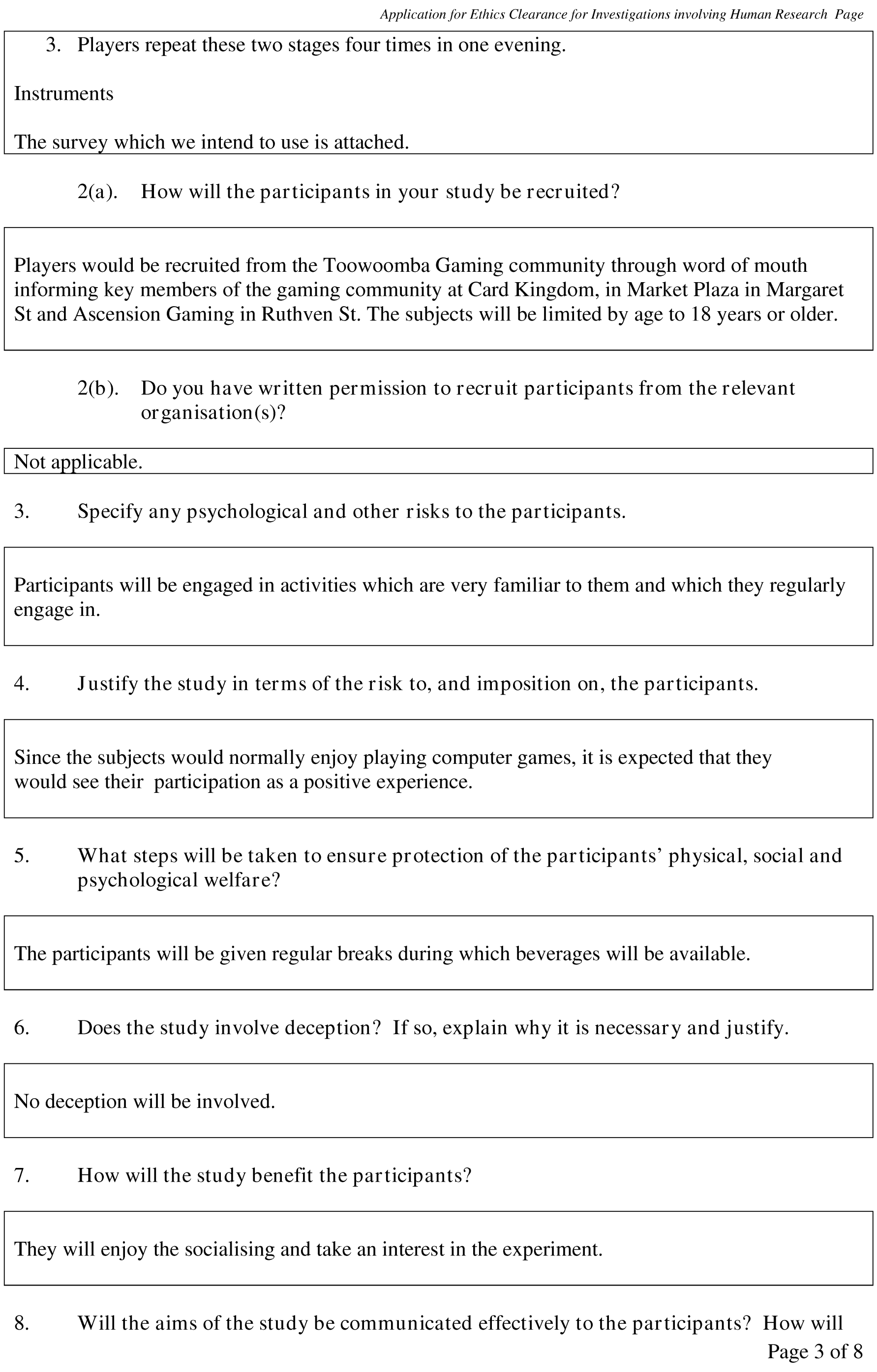
Experiment Description

- Participants play a computer game under changing computer/computer network performance to gather information on their gaming satisfaction perception.
- Every half hour approximately the participants will be asked to fill in a brief survey.
- The experiment would be over 3 evenings. Each evening, the experiment would take approximately 4 hours (6.30pm till 10.30pm). Participants only need to attend a single evening but are welcome to turn up for more sessions.
- Breaks would be taken regularly after each game session of approximately 30 minutes and participants would actually play the game for about half the time, taking the other half to relax and answer the Survey Questions.
- A survey (attached) will be filled in during breaks between game sessions.

2. Describe the study's stages, processes and instruments.

1. Participants will be asked to play a computer game for approximately half an hour (this may depend upon the style of computer game).
2. Participants fill in a survey (as follows) which seeks information about their experience while playing the computer game.

Figure C.2: USQ Ethics Clearance Page 2



Application for Ethics Clearance for Investigations involving Human Research Page

3. Players repeat these two stages four times in one evening.

Instruments

The survey which we intend to use is attached.

2(a). How will the participants in your study be recruited?

Players would be recruited from the Toowoomba Gaming community through word of mouth informing key members of the gaming community at Card Kingdom, in Market Plaza in Margaret St and Ascension Gaming in Ruthven St. The subjects will be limited by age to 18 years or older.

2(b). Do you have written permission to recruit participants from the relevant organisation(s)?

Not applicable.

3. Specify any psychological and other risks to the participants.

Participants will be engaged in activities which are very familiar to them and which they regularly engage in.

4. Justify the study in terms of the risk to, and imposition on, the participants.

Since the subjects would normally enjoy playing computer games, it is expected that they would see their participation as a positive experience.

5. What steps will be taken to ensure protection of the participants' physical, social and psychological welfare?

The participants will be given regular breaks during which beverages will be available.

6. Does the study involve deception? If so, explain why it is necessary and justify.

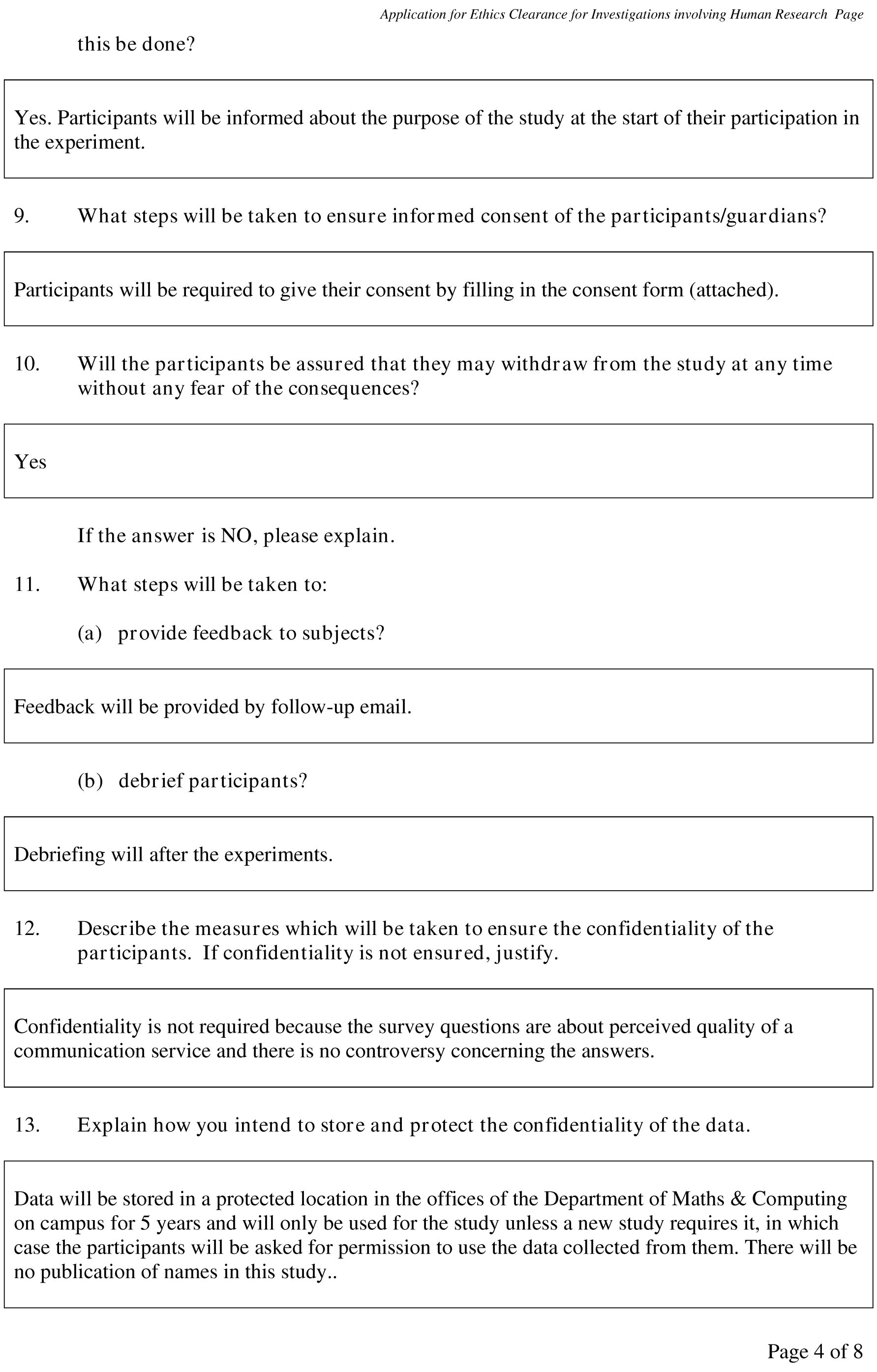
No deception will be involved.

7. How will the study benefit the participants?

They will enjoy the socialising and take an interest in the experiment.

8. Will the aims of the study be communicated effectively to the participants? How will

Figure C.3: USQ Ethics Clearance Page 3



this be done?

Yes. Participants will be informed about the purpose of the study at the start of their participation in the experiment.

9. What steps will be taken to ensure informed consent of the participants/guardians?

Participants will be required to give their consent by filling in the consent form (attached).

10. Will the participants be assured that they may withdraw from the study at any time without any fear of the consequences?

Yes

If the answer is NO, please explain.

11. What steps will be taken to:

(a) provide feedback to subjects?

Feedback will be provided by follow-up email.

(b) debrief participants?

Debriefing will after the experiments.

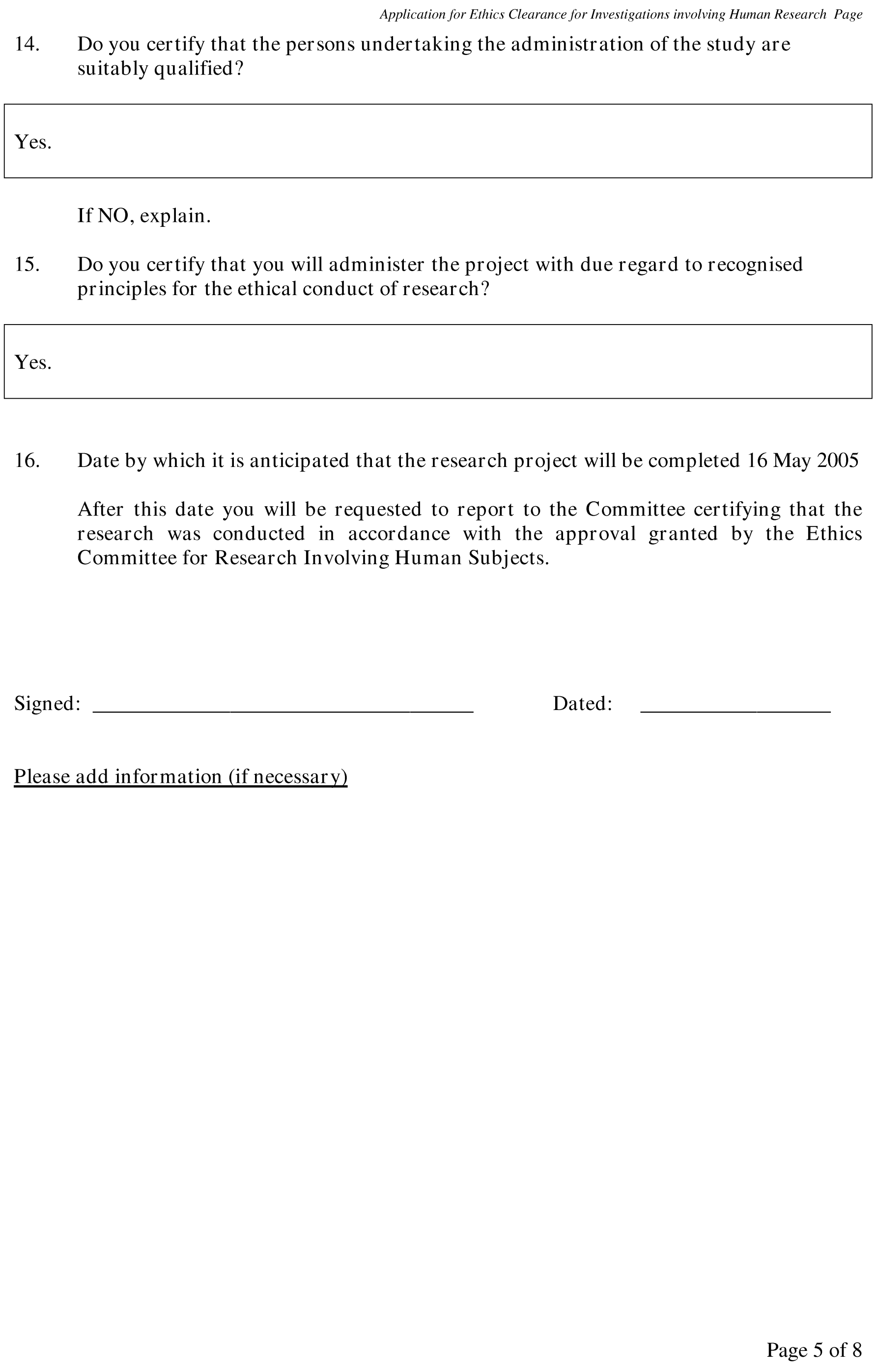
12. Describe the measures which will be taken to ensure the confidentiality of the participants. If confidentiality is not ensured, justify.

Confidentiality is not required because the survey questions are about perceived quality of a communication service and there is no controversy concerning the answers.

13. Explain how you intend to store and protect the confidentiality of the data.

Data will be stored in a protected location in the offices of the Department of Maths & Computing on campus for 5 years and will only be used for the study unless a new study requires it, in which case the participants will be asked for permission to use the data collected from them. There will be no publication of names in this study..

Figure C.4: USQ Ethics Clearance Page 4



14. Do you certify that the persons undertaking the administration of the study are suitably qualified?

Yes.

If NO, explain.

15. Do you certify that you will administer the project with due regard to recognised principles for the ethical conduct of research?

Yes.

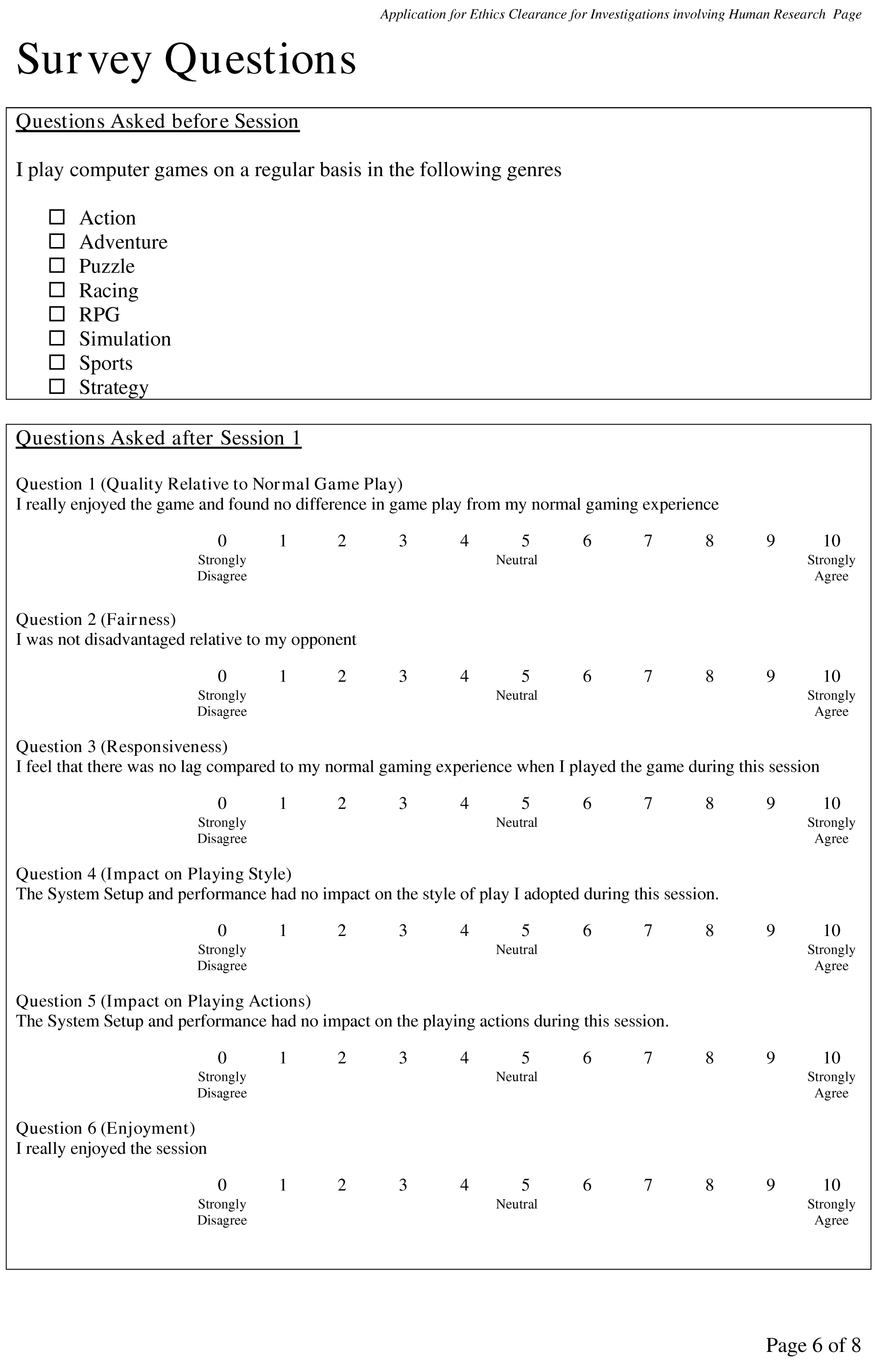
16. Date by which it is anticipated that the research project will be completed 16 May 2005

After this date you will be requested to report to the Committee certifying that the research was conducted in accordance with the approval granted by the Ethics Committee for Research Involving Human Subjects.

Signed: ___________________________________ Dated: _________________

Please add information (if necessary)

Figure C.5: USQ Ethics Clearance Page 5



Survey Questions

Questions Asked before Session

I play computer games on a regular basis in the following genres

- □ Action
- □ Adventure
- □ Puzzle
- □ Racing
- □ RPG
- □ Simulation
- □ Sports
- □ Strategy

Questions Asked after Session 1

Question 1 (Quality Relative to Normal Game Play)
I really enjoyed the game and found no difference in game play from my normal gaming experience

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree					Neutral					Strongly Agree

Question 2 (Fairness)
I was not disadvantaged relative to my opponent

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree					Neutral					Strongly Agree

Question 3 (Responsiveness)
I feel that there was no lag compared to my normal gaming experience when I played the game during this session

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree					Neutral					Strongly Agree

Question 4 (Impact on Playing Style)
The System Setup and performance had no impact on the style of play I adopted during this session.

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree					Neutral					Strongly Agree

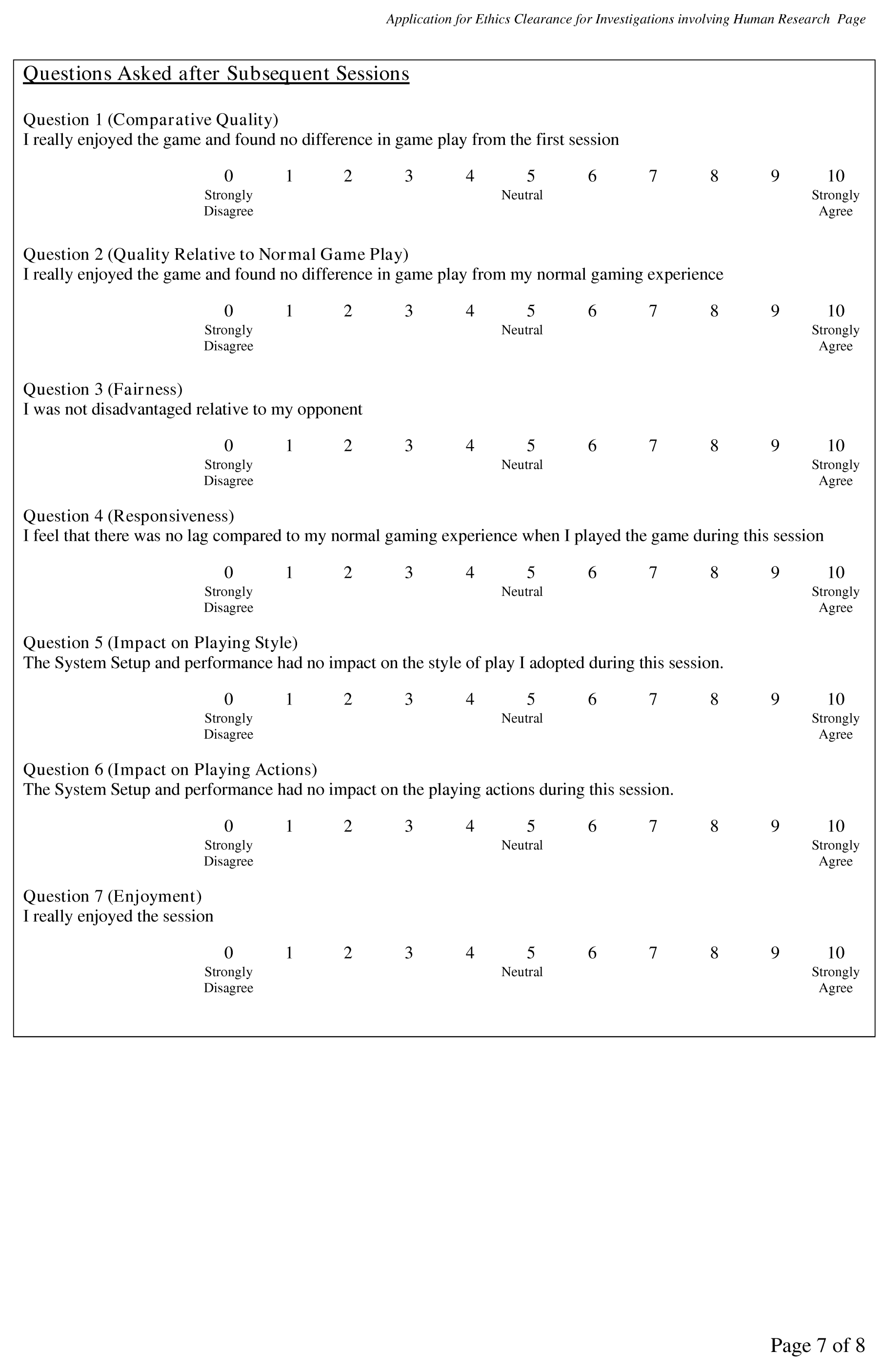
Question 5 (Impact on Playing Actions)
The System Setup and performance had no impact on the playing actions during this session.

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree					Neutral					Strongly Agree

Question 6 (Enjoyment)
I really enjoyed the session

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree					Neutral					Strongly Agree

Figure C.6: USQ Ethics Clearance Page 6



<u>Questions Asked after Subsequent Sessions</u>

Question 1 (Comparative Quality)
I really enjoyed the game and found no difference in game play from the first session

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree					Neutral					Strongly Agree

Question 2 (Quality Relative to Normal Game Play)
I really enjoyed the game and found no difference in game play from my normal gaming experience

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree					Neutral					Strongly Agree

Question 3 (Fairness)
I was not disadvantaged relative to my opponent

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree					Neutral					Strongly Agree

Question 4 (Responsiveness)
I feel that there was no lag compared to my normal gaming experience when I played the game during this session

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree					Neutral					Strongly Agree

Question 5 (Impact on Playing Style)
The System Setup and performance had no impact on the style of play I adopted during this session.

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree					Neutral					Strongly Agree

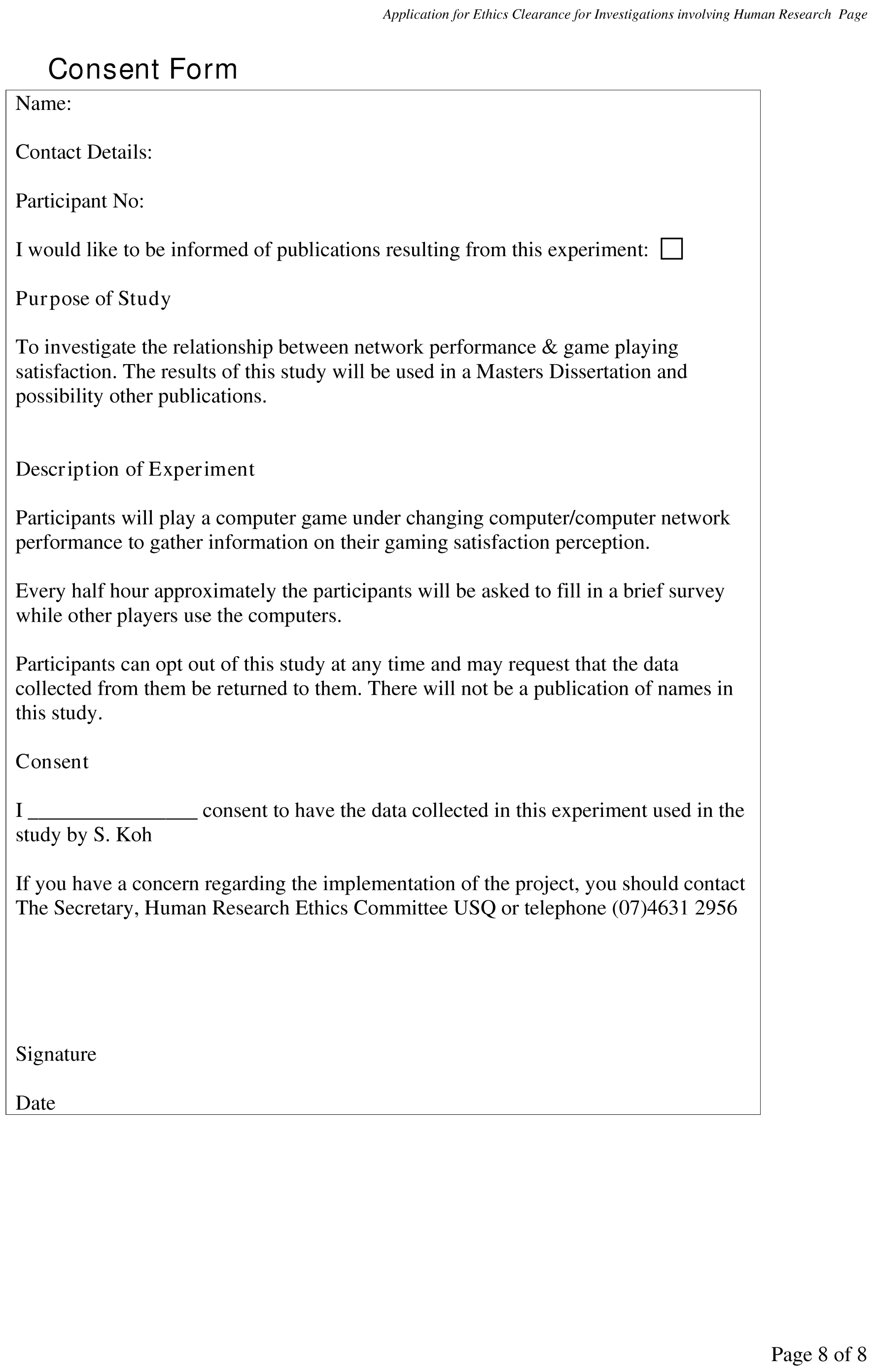
Question 6 (Impact on Playing Actions)
The System Setup and performance had no impact on the playing actions during this session.

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree					Neutral					Strongly Agree

Question 7 (Enjoyment)
I really enjoyed the session

0	1	2	3	4	5	6	7	8	9	10
Strongly Disagree					Neutral					Strongly Agree

Figure C.7: USQ Ethics Clearance Page 7



Application for Ethics Clearance for Investigations involving Human Research Page

Consent Form

Name:

Contact Details:

Participant No:

I would like to be informed of publications resulting from this experiment: □

Purpose of Study

To investigate the relationship between network performance & game playing satisfaction. The results of this study will be used in a Masters Dissertation and possibility other publications.

Description of Experiment

Participants will play a computer game under changing computer/computer network performance to gather information on their gaming satisfaction perception.

Every half hour approximately the participants will be asked to fill in a brief survey while other players use the computers.

Participants can opt out of this study at any time and may request that the data collected from them be returned to them. There will not be a publication of names in this study.

Consent

I ________________ consent to have the data collected in this experiment used in the study by S. Koh

If you have a concern regarding the implementation of the project, you should contact The Secretary, Human Research Ethics Committee USQ or telephone (07)4631 2956

Signature

Date

Page 8 of 8

Figure C.8: USQ Ethics Clearance Page 8

Appendix D

Network Traffic Plots

D.1 Age of Mythology

D.1.1 Session 1

Game	Age of Mythology
Session	1
Nist.Net Delay Setting	0
Nist.Net Bandwidth Setting	12500000
Nist.Net Drop % Setting	0
Nist.Net Duplication % Setting	0
Total Elapsed Time	3660
Total Number of Packets	68406
Total Bytes Transfer	2770210
Average Number of Packets per Second	18.68505873
Average Bytes Transferred per Second	756.6812346
Average Size per Packet	40.49659387
Total Number of UDP Packets	68357
Total Size of UDP Data	2766728
Average Number of UDP Packets per Second	18.67167441
Average Size of UDP Transferred per Second	755.7301284
Average size per UDP Packet	40.47468438
Total Number of TCP Packets	34
Total Size of TCP Data	2582
Average Number of TCP Packets per Second	0.00928708
Average Size of TCP Transferred per Second	0.705271784

Average size per TCP Packet	75.94117647
Total Number of OTHERS Packets	15
Total Size of OTHERS Data	900
Average Number of OTHERS Packets per Second	0.004097241
Average Size of OTHERS Transferred per Second	0.245834471
Average size per OTHERS Packet	60

Traffic Graph

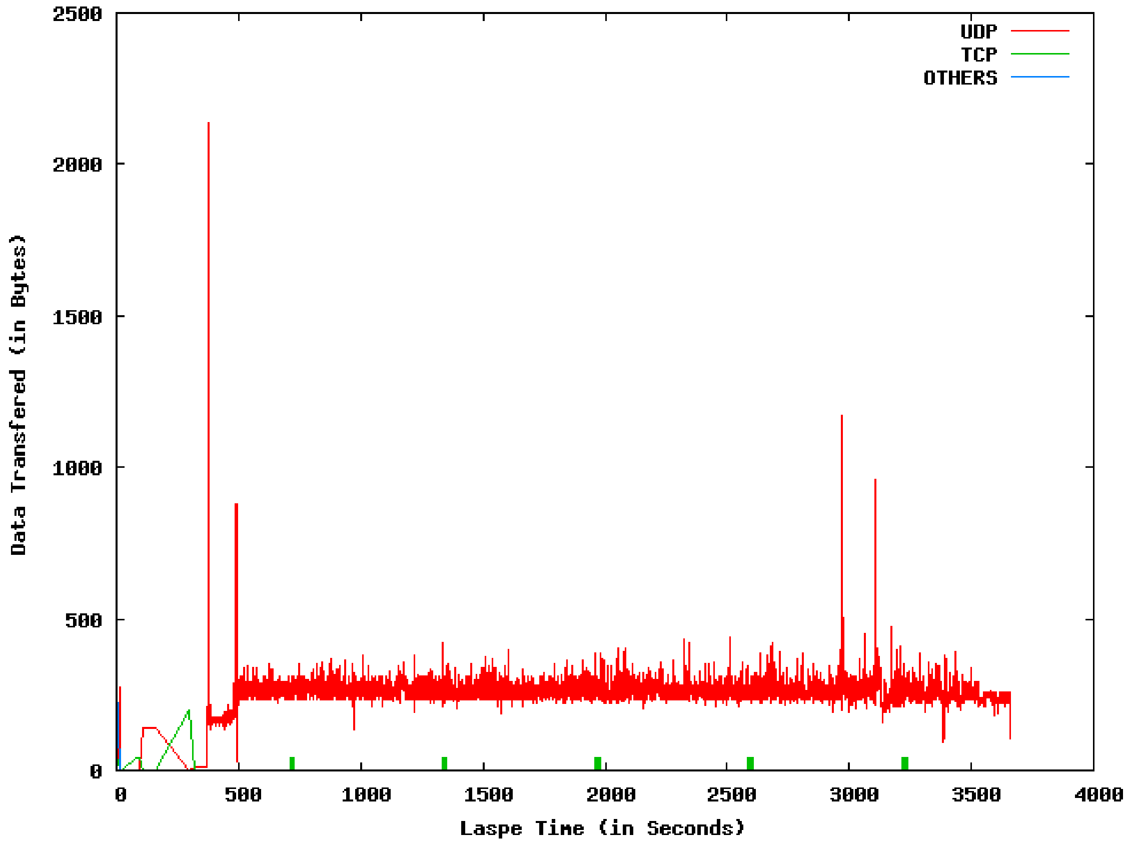


Figure D.1: Age of Mythology Traffic Chart

Packets Graph

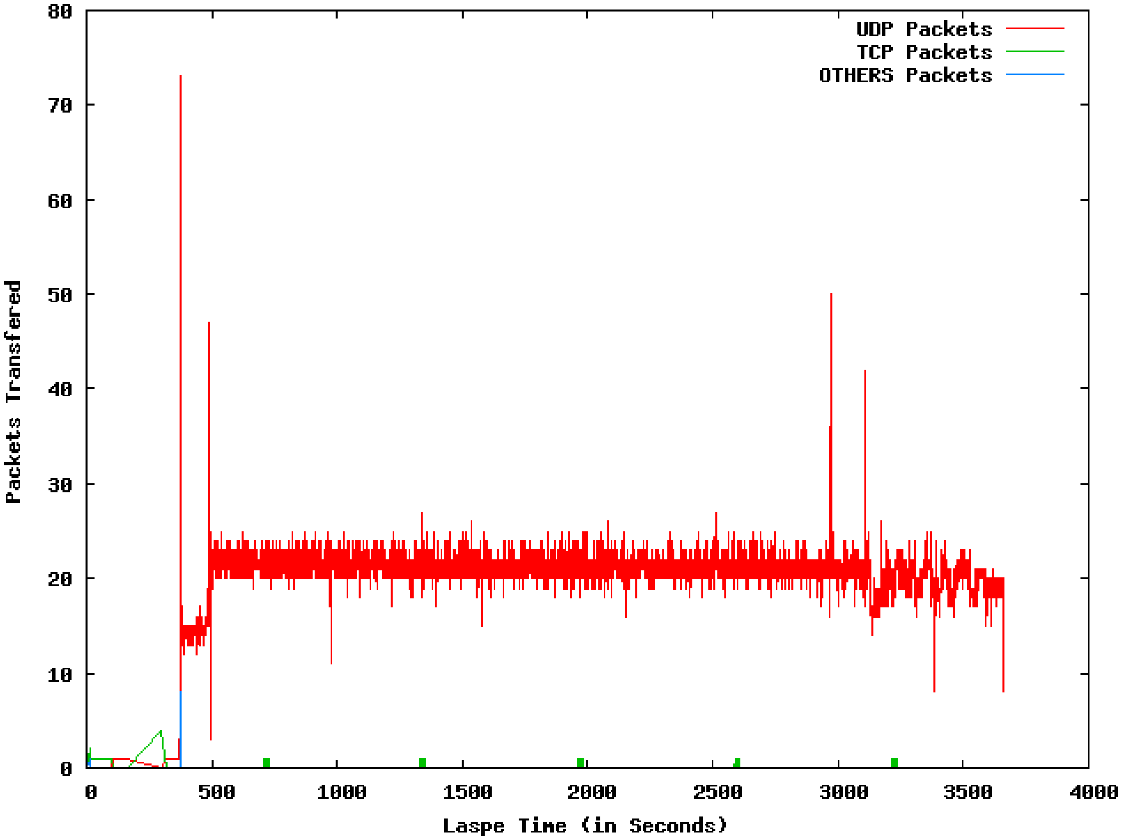


Figure D.2: Age of Mythology Packets Chart

D.1.2 Session 2

Game	Age of Mythology
Session	2
Nist.Net Delay Setting	200
Nist.Net Bandwidth Setting	9600
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	0
Total Elapsed Time	3290
Total Number of Packets	86017
Total Bytes Transfer	3636982
Average Number of Packets per Second	26.13704041
Average Bytes Transferred per Second	1105.129748
Average Size per Packet	42.28213028
Total Number of UDP Packets	85766
Total Size of UDP Data	3621495
Average Number of UDP Packets per Second	26.0607718
Average Size of UDP Transferred per Second	1100.423883
Average size per UDP Packet	42.22529907
Total Number of TCP Packets	211
Total Size of TCP Data	11087
Average Number of TCP Packets per Second	0.064114251
Average Size of TCP Transferred per Second	3.368884837
Average size per TCP Packet	52.5450237
Total Number of OTHERS Packets	40
Total Size of OTHERS Data	4400
Average Number of OTHERS Packets per Second	0.01215436
Average Size of OTHERS Transferred per Second	1.336979641
Average size per OTHERS Packet	110

Traffic Graph

Packets Graph

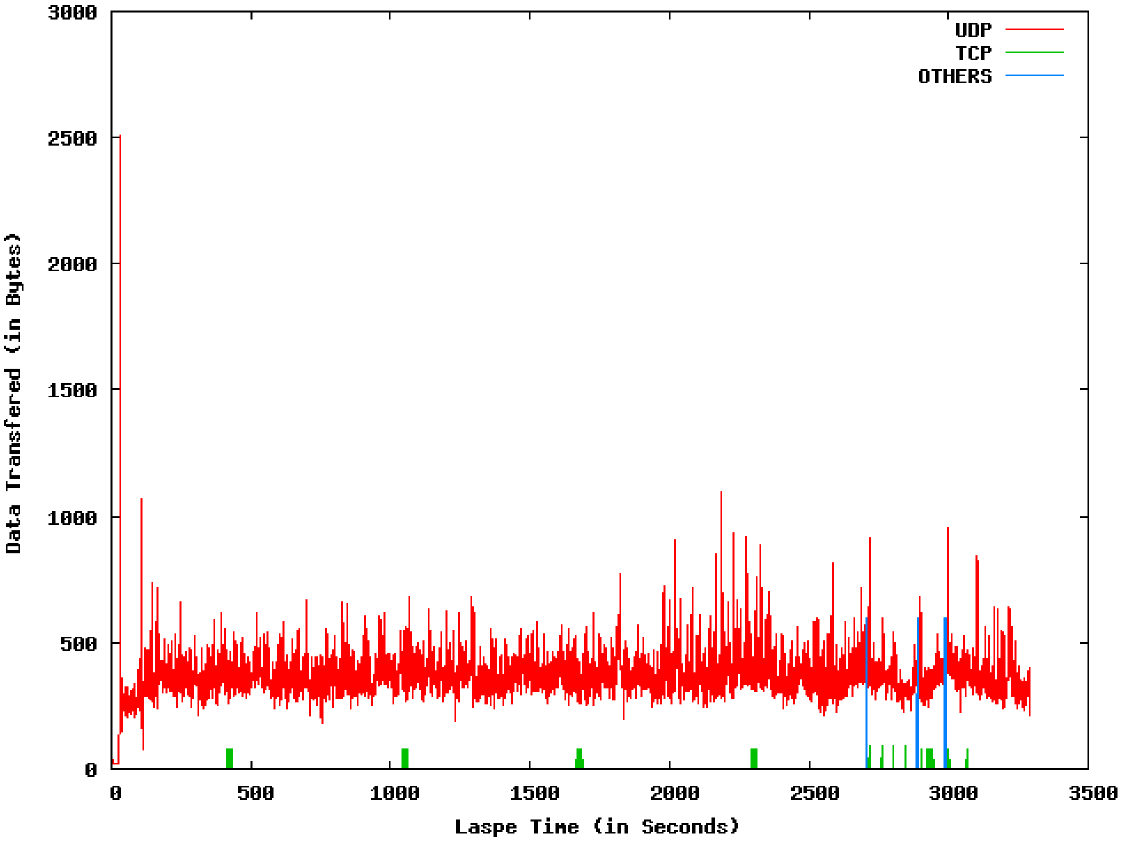


Figure D.3: Age of Mythology Traffic Chart

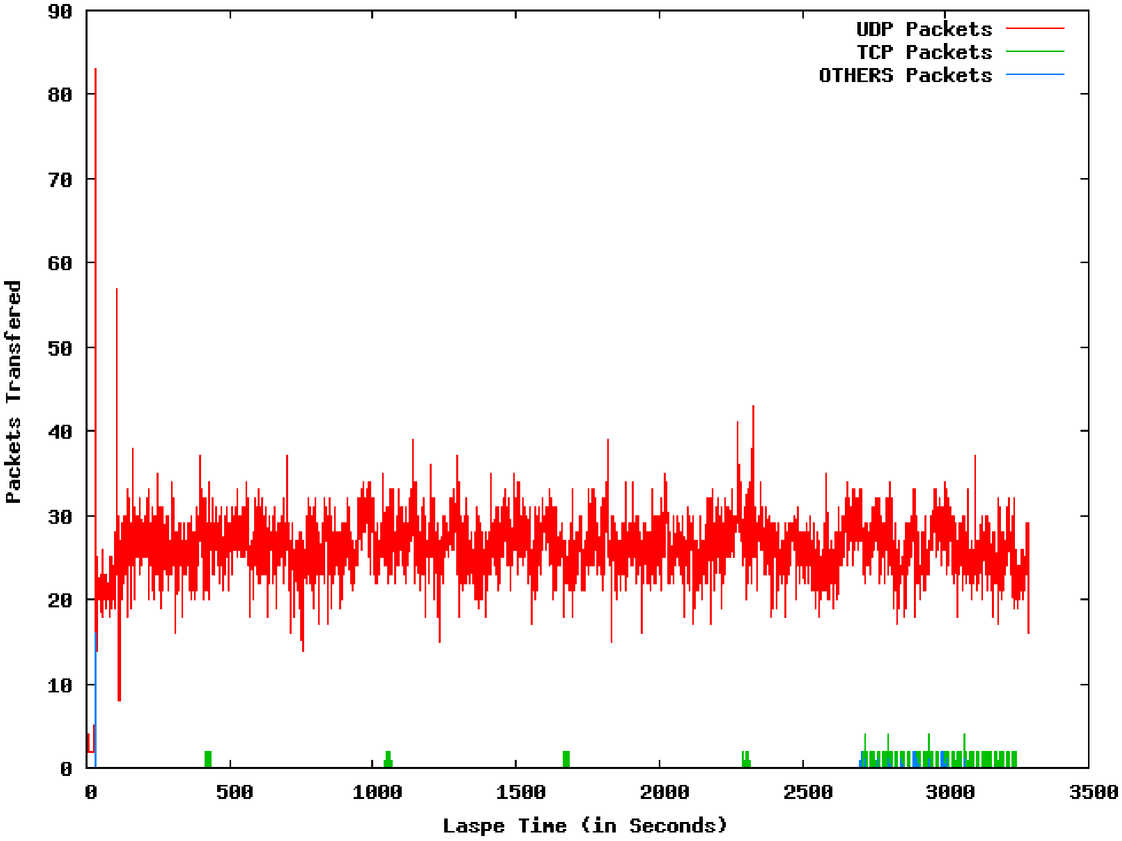


Figure D.4: Age of Mythology Packets Chart

D.1.3 Session 3

Game	Age of Mythology
Session	3
Nist.Net Delay Setting	100
Nist.Net Bandwidth Setting	500
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	0
Total Elapsed Time	2556
Total Number of Packets	63656
Total Bytes Transfer	2644242
Average Number of Packets per Second	24.00
Average Bytes Transferred per Second	1034.00
Average Size per Packet	41.00
Total Number of UDP Packets	63608
Total Size of UDP Data	2640998
Average Number of UDP Packets per Second	24.00
Average Size of UDP Transferred per Second	1032.00
Average size per UDP Packet	41.00
Total Number of TCP Packets	38
Total Size of TCP Data	3044
Average Number of TCP Packets per Second	0.00
Average Size of TCP Transferred per Second	1.00
Average size per TCP Packet	80.00
Total Number of OTHERS Packets	10
Total Size of OTHERS Data	200
Average Number of OTHERS Packets per Second	0.00
Average Size of OTHERS Transferred per Second	0.00
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

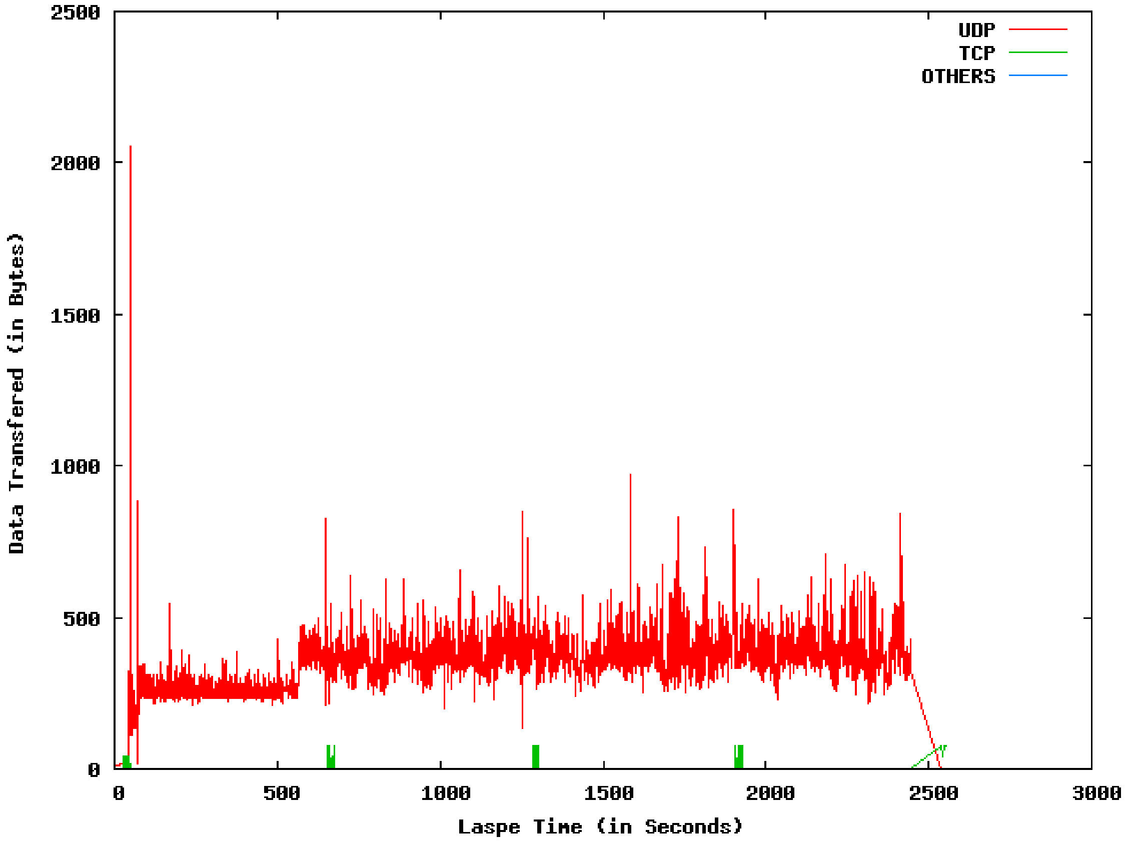


Figure D.5: Age of Mythology Traffic Chart

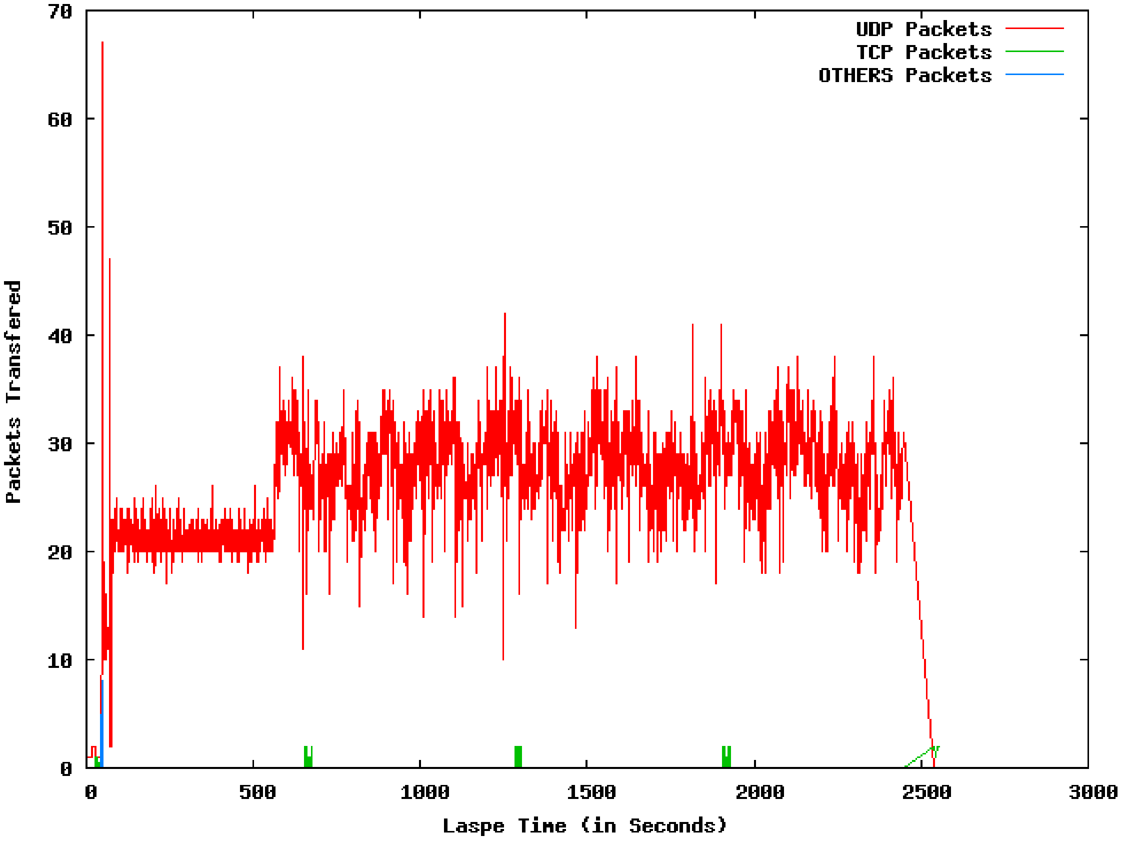


Figure D.6: Age of Mythology Packets Chart

D.1.4 Session 4

Game	Age of Mythology
Session	4
Nist.Net Delay Setting	500
Nist.Net Bandwidth Setting	1200
Nist.Net Drop % Setting	10
Nist.Net Duplication % Setting	5
Total Elapsed Time	3722
Total Number of Packets	80176
Total Bytes Transfer	3531520
Average Number of Packets per Second	21.53532098
Average Bytes Transferred per Second	948.5683589
Average Size per Packet	44.04709639
Total Number of UDP Packets	80104
Total Size of UDP Data	3527366
Average Number of UDP Packets per Second	21.51598174
Average Size of UDP Transferred per Second	947.452592
Average size per UDP Packet	44.03482972
Total Number of TCP Packets	46
Total Size of TCP Data	3634
Average Number of TCP Packets per Second	0.012355627
Average Size of TCP Transferred per Second	0.976094547
Average size per TCP Packet	79
Total Number of OTHERS Packets	26
Total Size of OTHERS Data	520
Average Number of OTHERS Packets per Second	0.006983615
Average Size of OTHERS Transferred per Second	0.139672307
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

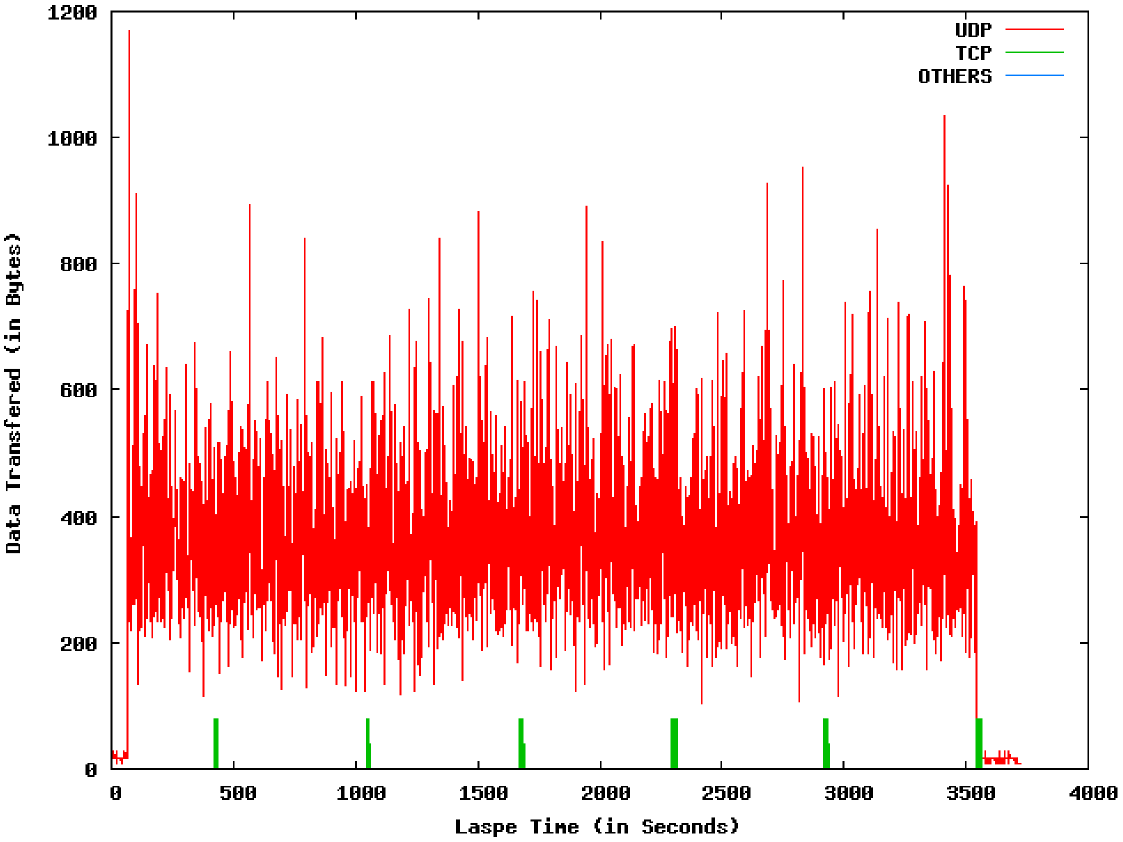


Figure D.7: Age of Mythology Traffic Chart

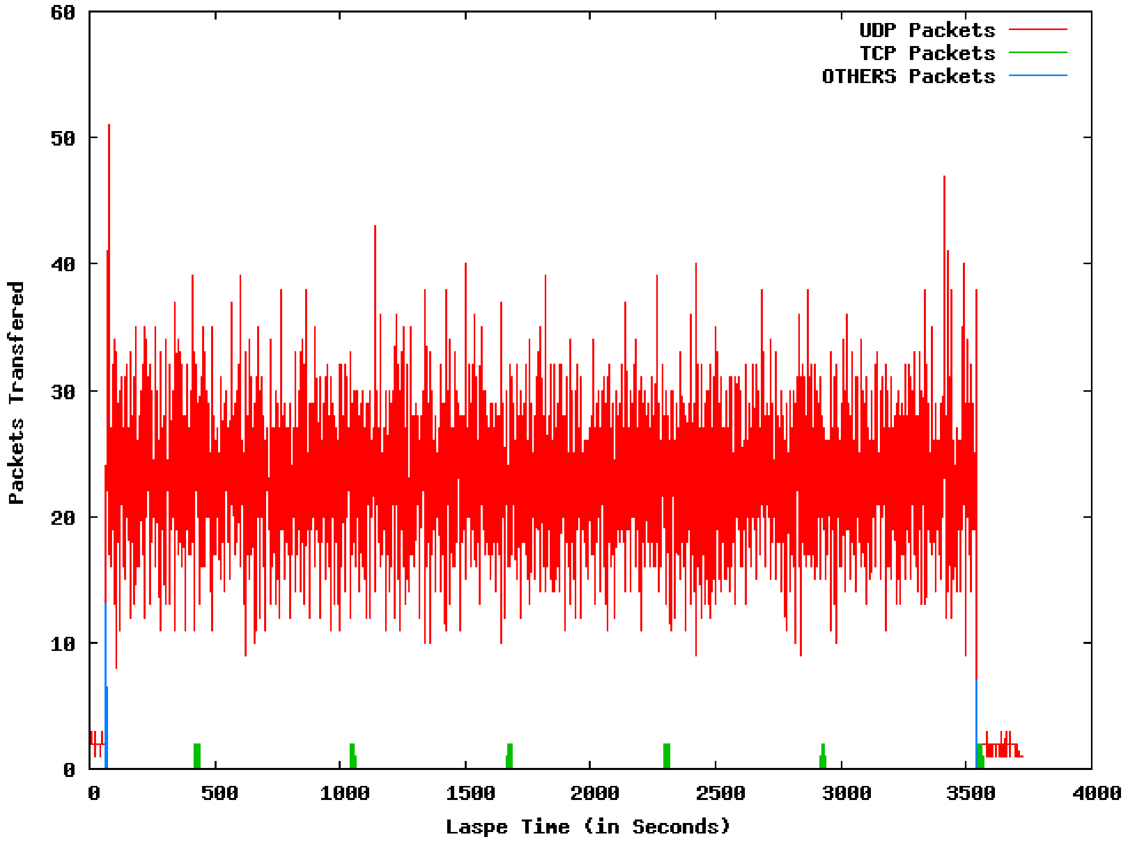


Figure D.8: Age of Mythology Packets Chart

D.1.5 Session 5

Game	Age of Mythology
Session	5
Nist.Net Delay Setting	50
Nist.Net Bandwidth Setting	500
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	5
Total Elapsed Time	1804
Total Number of Packets	30909
Total Bytes Transfer	1247734
Average Number of Packets per Second	17.12409972
Average Bytes Transferred per Second	691.265374
Average Size per Packet	40.36798344
Total Number of UDP Packets	30887
Total Size of UDP Data	1246586
Average Number of UDP Packets per Second	17.11191136
Average Size of UDP Transferred per Second	690.6293629
Average size per UDP Packet	40.35956875
Total Number of TCP Packets	12
Total Size of TCP Data	948
Average Number of TCP Packets per Second	0.006648199
Average Size of TCP Transferred per Second	0.525207756
Average size per TCP Packet	79
Total Number of OTHERS Packets	10
Total Size of OTHERS Data	200
Average Number of OTHERS Packets per Second	0.005540166
Average Size of OTHERS Transferred per Second	0.110803324
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

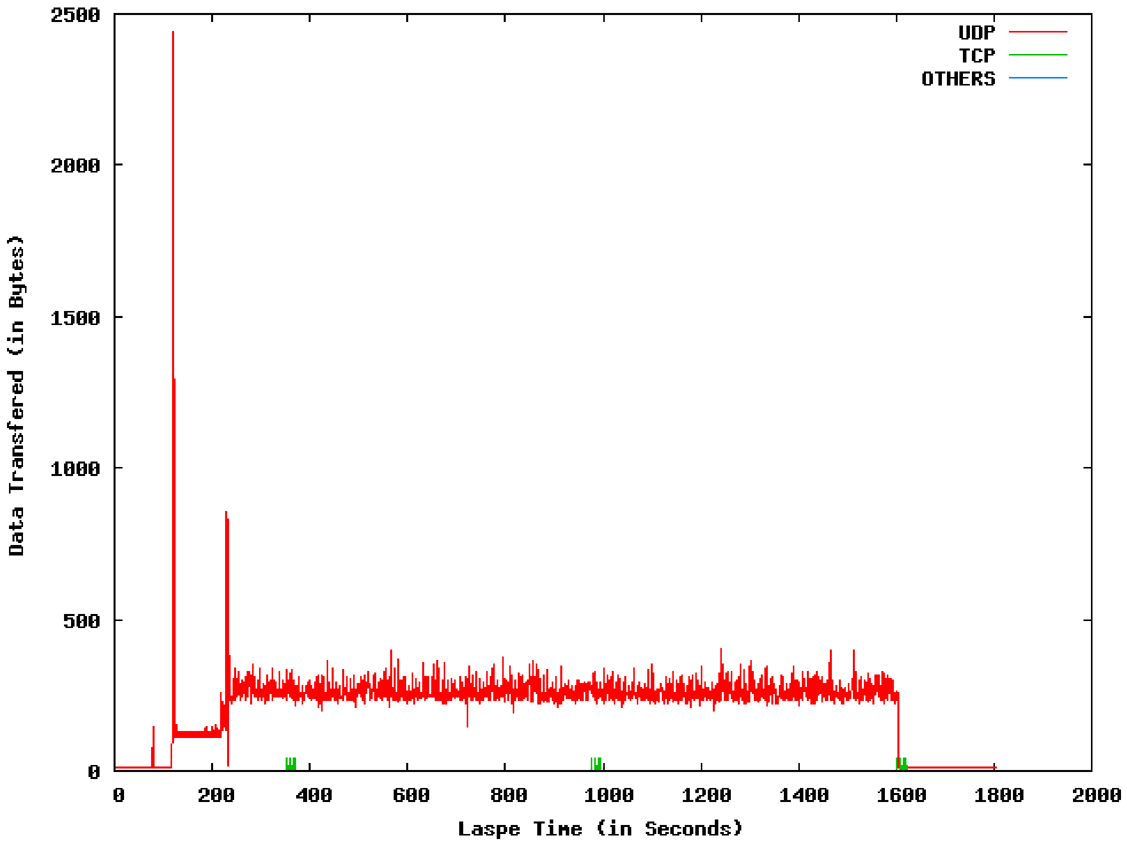


Figure D.9: Age of Mythology Traffic Chart

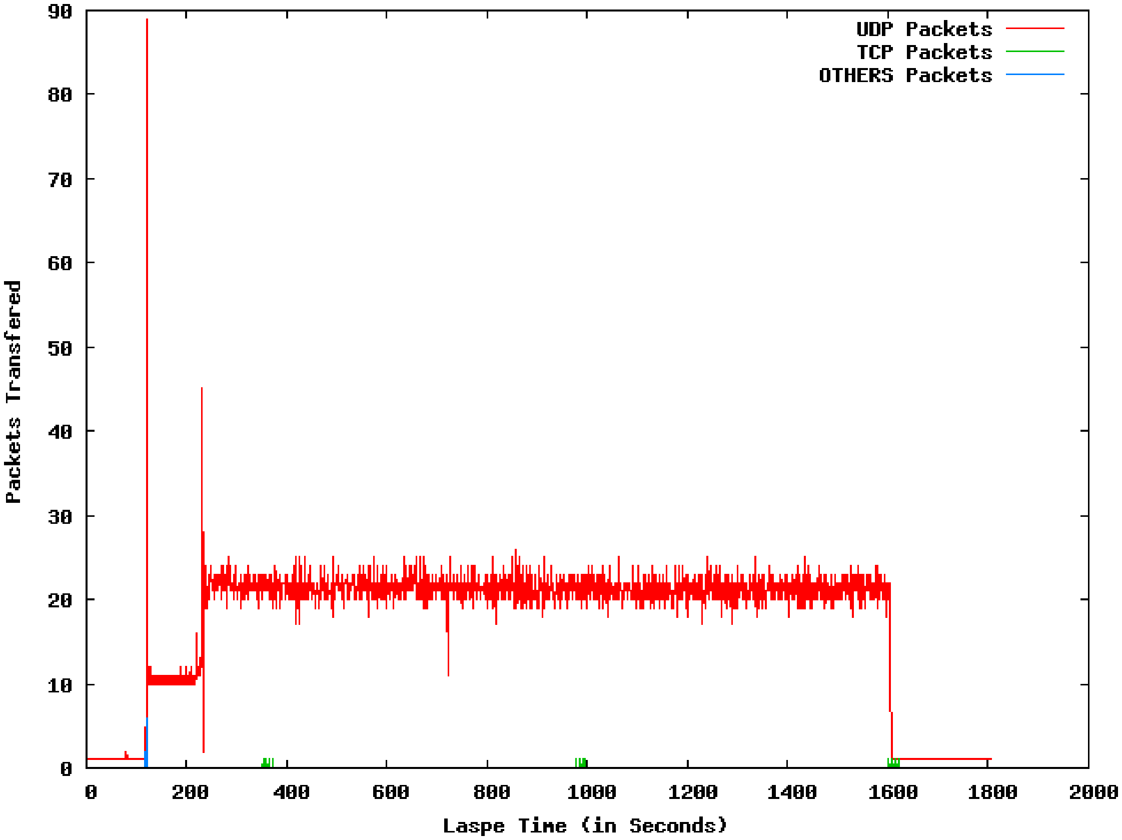


Figure D.10: Age of Mythology Packets Chart

D.2 Battlefield 1942

D.2.1 Session 1

Game	**Battlefield 1942**
Session	1
Nist.Net Delay Setting	0
Nist.Net Bandwidth Setting	12500000
Nist.Net Drop % Setting	0
Nist.Net Duplication % Setting	0
Total Elapsed Time	723
Total Number of Packets	17045
Total Bytes Transfer	1735072
Average Number of Packets per Second	23.542817679558
Average Bytes Transferred per Second	2396.50828729282
Average Size per Packet	101.793605162804
Total Number of UDP Packets	16980
Total Size of UDP Data	1732874
Average Number of UDP Packets per Second	23.4530386740331
Average Size of UDP Transferred per Second	2393.47237569061
Average size per UDP Packet	102.05382803298
Total Number of TCP Packets	21
Total Size of TCP Data	1318
Average Number of TCP Packets per Second	0.0290055248618785
Average Size of TCP Transferred per Second	1.82044198895028
Average size per TCP Packet	62.7619047619048
Total Number of OTHERS Packets	44
Total Size of OTHERS Data	880
Average Number of OTHERS Packets per Second	0.0607734806629834
Average Size of OTHERS Transferred per Second	1.21546961325967
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

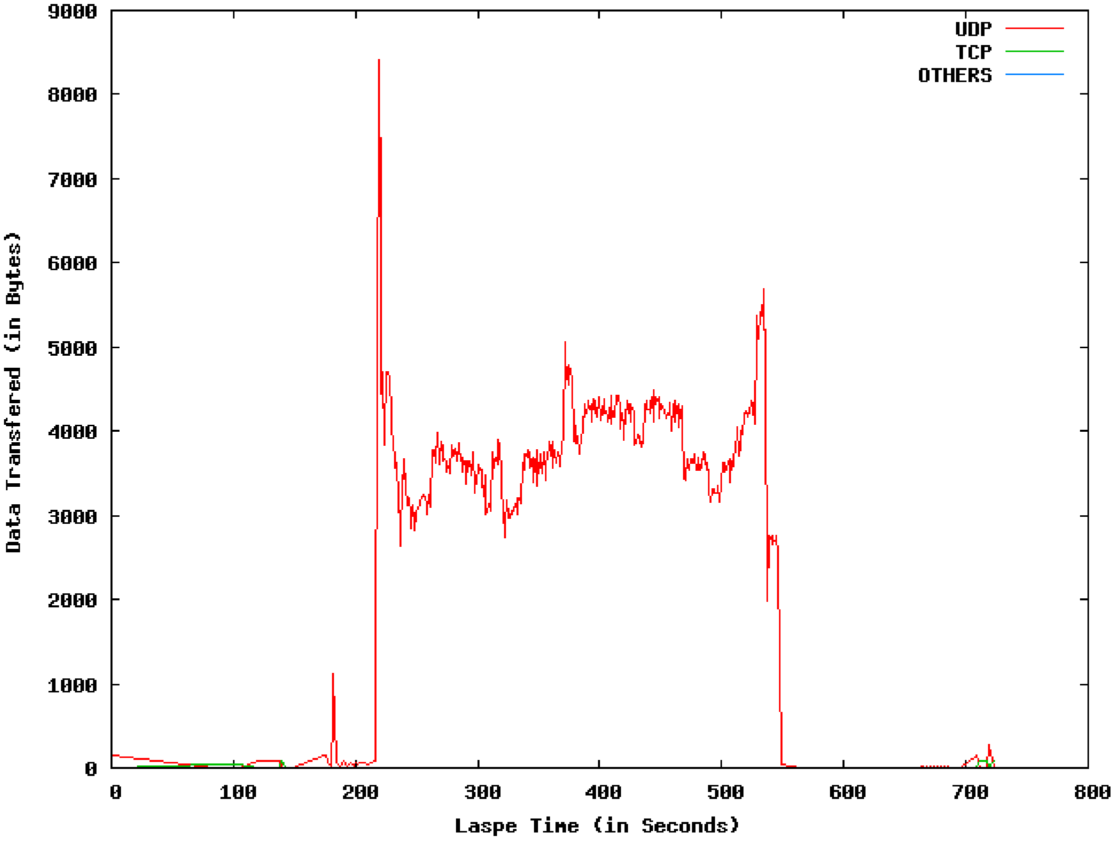


Figure D.11: Battlefield 1942 Traffic Chart

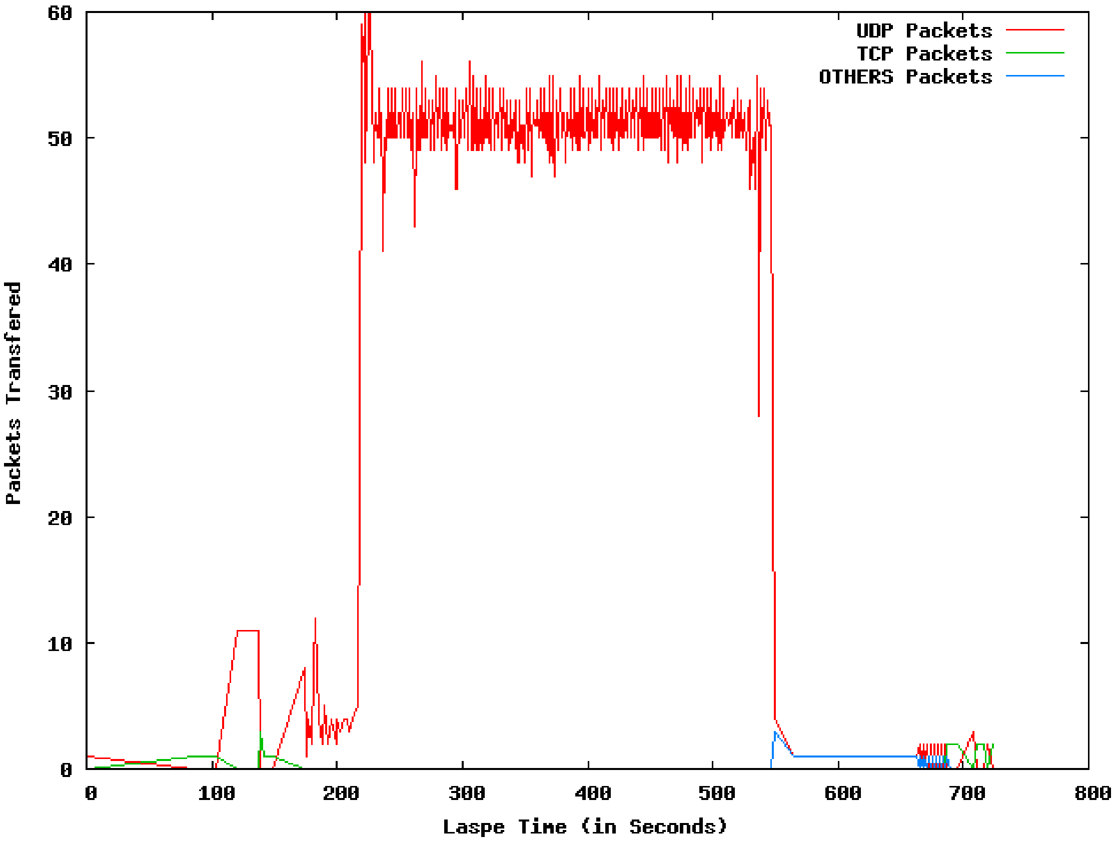


Figure D.12: Battlefield 1942 Packets Chart

D.2.2 Session 2

Game	**Battlefield 1942**
Session	2
Nist.Net Delay Setting	200
Nist.Net Bandwidth Setting	9600
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	0
Total Elapsed Time	359
Total Number of Packets	24627
Total Bytes Transfer	2383359
Average Number of Packets per Second	68.4083333333333
Average Bytes Transferred per Second	6620.44166666667
Average Size per Packet	96.7782921184066
Total Number of UDP Packets	24618
Total Size of UDP Data	2382941
Average Number of UDP Packets per Second	68.3833333333333
Average Size of UDP Transferred per Second	6619.28055555556
Average size per UDP Packet	96.7966934763181
Total Number of TCP Packets	8
Total Size of TCP Data	398
Average Number of TCP Packets per Second	0.0222222222222222
Average Size of TCP Transferred per Second	1.10555555555556
Average size per TCP Packet	49.75
Total Number of OTHERS Packets	1
Total Size of OTHERS Data	20
Average Number of OTHERS Packets per Second	0.00277777777777778
Average Size of OTHERS Transferred per Second	0.0555555555555556
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

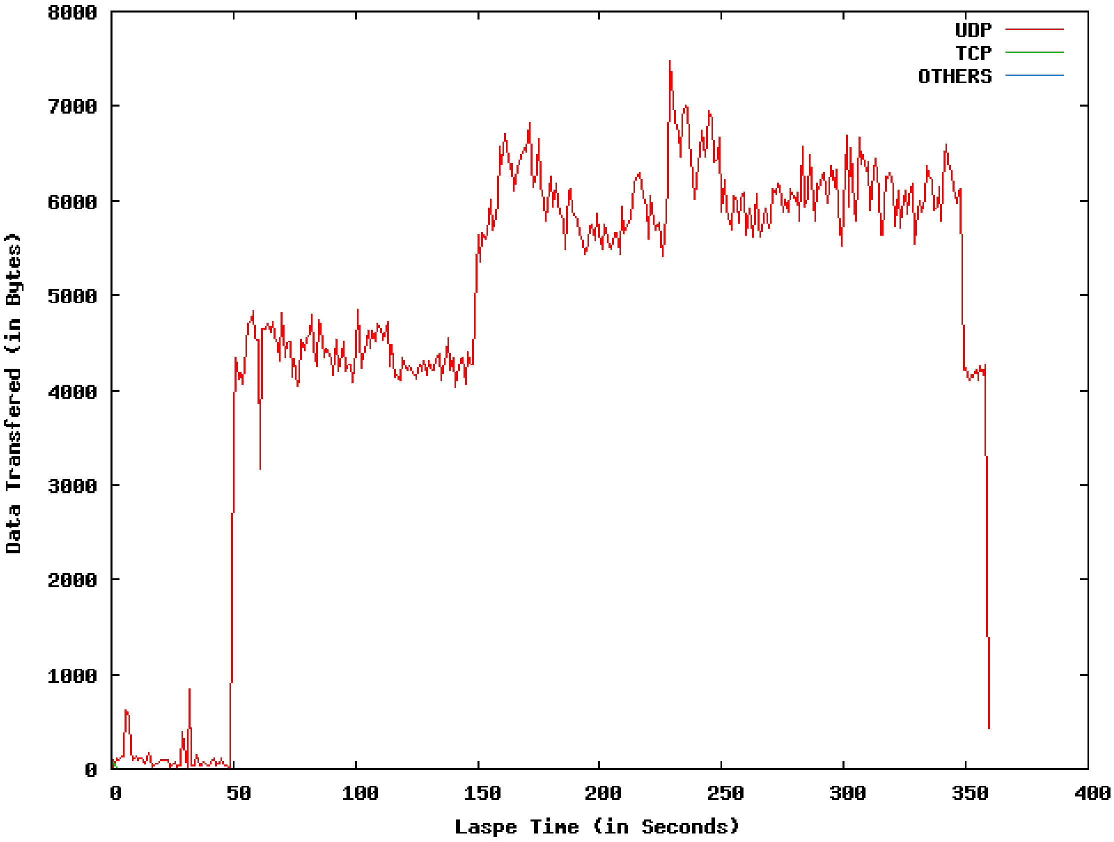


Figure D.13: Battlefield 1942 Traffic Chart

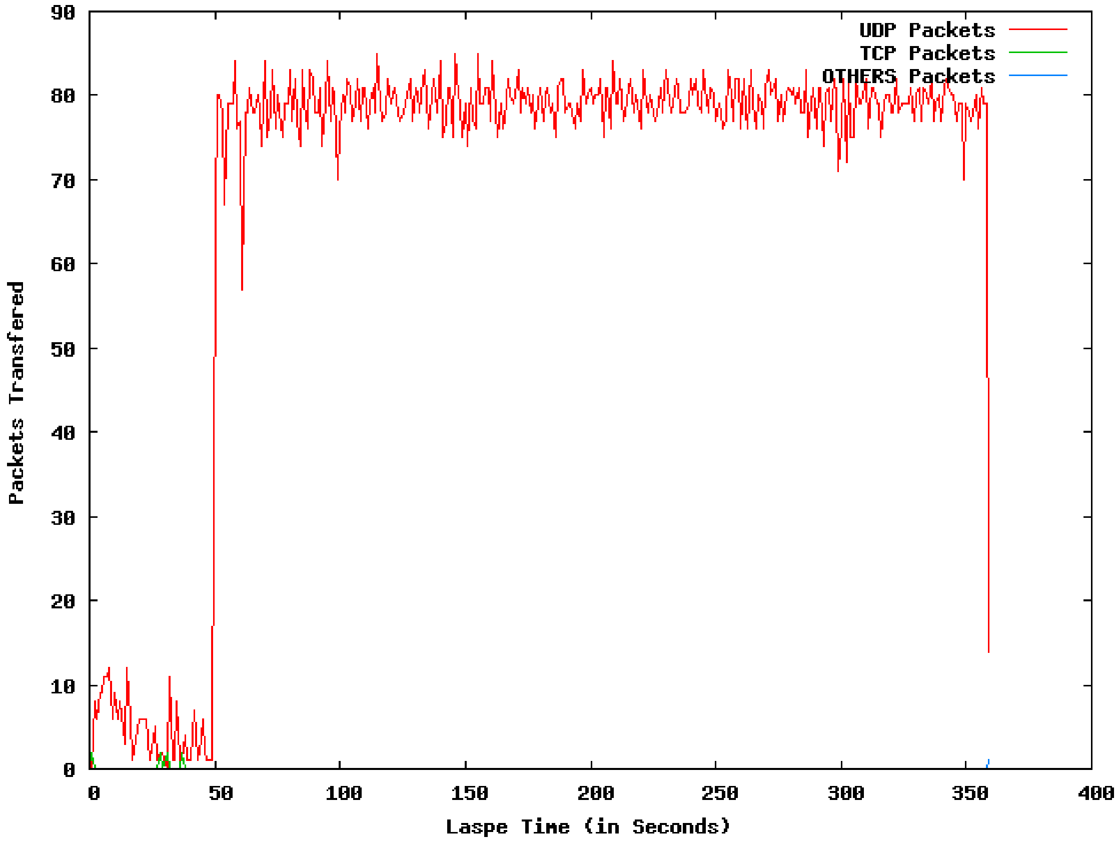


Figure D.14: Battlefield 1942 Packets Chart

D.2.3 Session 3

Game	**Battlefield 1942**
Session	3
Nist.Net Delay Setting	500
Nist.Net Bandwidth Setting	1000
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	0
Total Elapsed Time	230
Total Number of Packets	982
Total Bytes Transfer	62331
Average Number of Packets per Second	4.251082251
Average Bytes Transferred per Second	269.8311688
Average Size per Packet	63.47352342
Total Number of UDP Packets	830
Total Size of UDP Data	58635
Average Number of UDP Packets per Second	3.593073593
Average Size of UDP Transferred per Second	253.8311688
Average size per UDP Packet	70.64457831
Total Number of TCP Packets	25
Total Size of TCP Data	1156
Average Number of TCP Packets per Second	0.108225108
Average Size of TCP Transferred per Second	5.004329004
Average size per TCP Packet	46.24
Total Number of OTHERS Packets	127
Total Size of OTHERS Data	2540
Average Number of OTHERS Packets per Second	0.54978355
Average Size of OTHERS Transferred per Second	10.995671
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

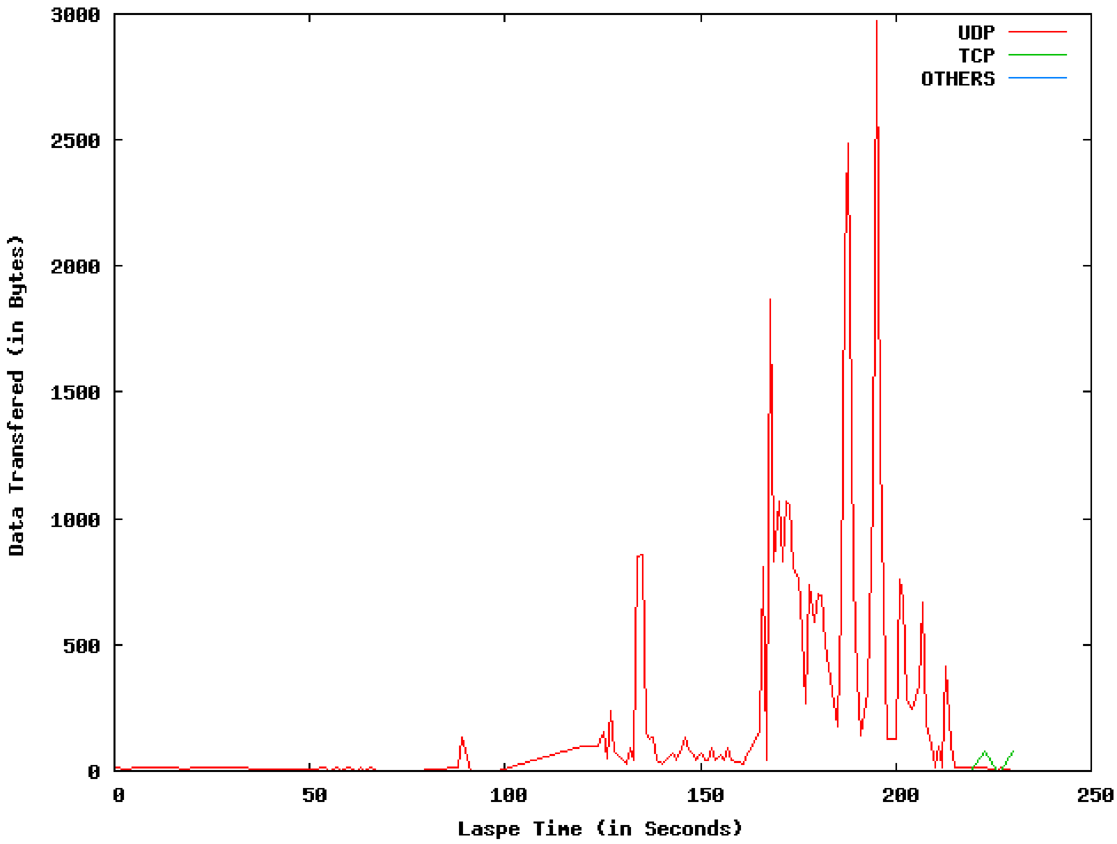


Figure D.15: Battlefield 1942 Traffic Chart

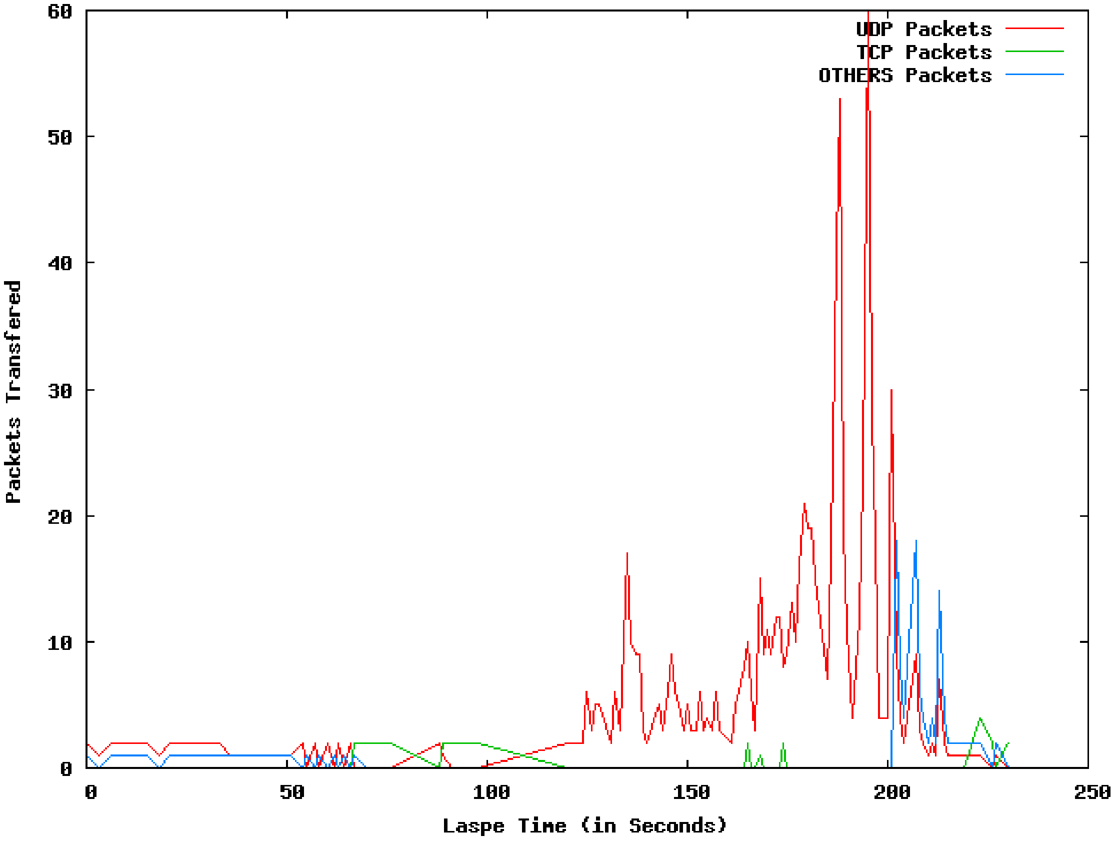


Figure D.16: Battlefield 1942 Packets Chart

D.2.4 Session 4

Game	**Battlefield 1942**
Session	4
Nist.Net Delay Setting	200
Nist.Net Bandwidth Setting	1000
Nist.Net Drop % Setting	10
Nist.Net Duplication % Setting	5
Total Elapsed Time	178
Total Number of Packets	616
Total Bytes Transfer	55742
Average Number of Packets per Second	3.441340782
Average Bytes Transferred per Second	311.4078212
Average Size per Packet	90.49025974
Total Number of UDP Packets	530
Total Size of UDP Data	53542
Average Number of UDP Packets per Second	2.960893855
Average Size of UDP Transferred per Second	299.1173184
Average size per UDP Packet	101.0226415
Total Number of TCP Packets	24
Total Size of TCP Data	960
Average Number of TCP Packets per Second	0.134078212
Average Size of TCP Transferred per Second	5.363128492
Average size per TCP Packet	40
Total Number of OTHERS Packets	62
Total Size of OTHERS Data	1240
Average Number of OTHERS Packets per Second	0.346368715
Average Size of OTHERS Transferred per Second	6.927374302
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

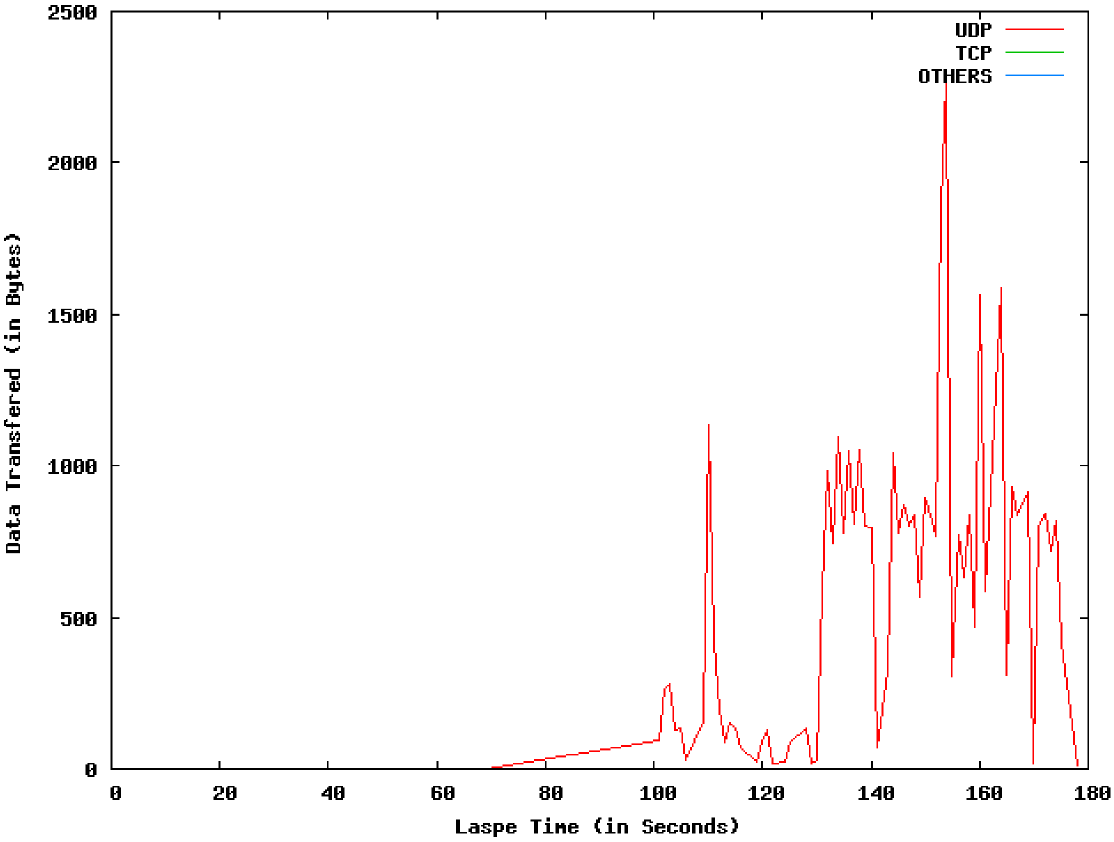


Figure D.17: Battlefield 1942 Traffic Chart

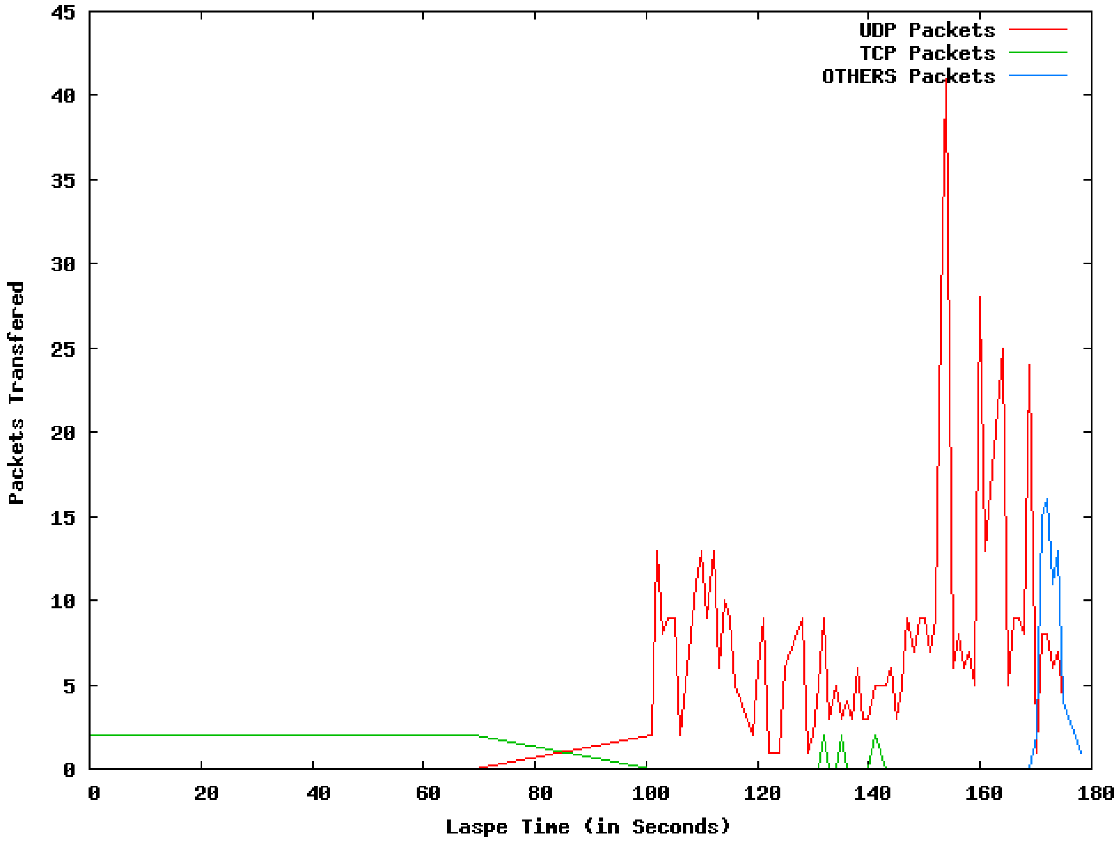


Figure D.18: Battlefield 1942 Packets Chart

D.2.5 Session 5

Game	**Battlefield 1942**
Session	5
Nist.Net Delay Setting	200
Nist.Net Bandwidth Setting	5000
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	5
Total Elapsed Time	405
Total Number of Packets	19863
Total Bytes Transfer	2071994
Average Number of Packets per Second	48.92364532
Average Bytes Transferred per Second	5103.433498
Average Size per Packet	104.3142526
Total Number of UDP Packets	19853
Total Size of UDP Data	2071674
Average Number of UDP Packets per Second	48.89901478
Average Size of UDP Transferred per Second	5102.64532
Average size per UDP Packet	104.3506775
Total Number of TCP Packets	6
Total Size of TCP Data	240
Average Number of TCP Packets per Second	0.014778325
Average Size of TCP Transferred per Second	0.591133005
Average size per TCP Packet	40
Total Number of OTHERS Packets	4
Total Size of OTHERS Data	80
Average Number of OTHERS Packets per Second	0.009852217
Average Size of OTHERS Transferred per Second	0.197044335
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

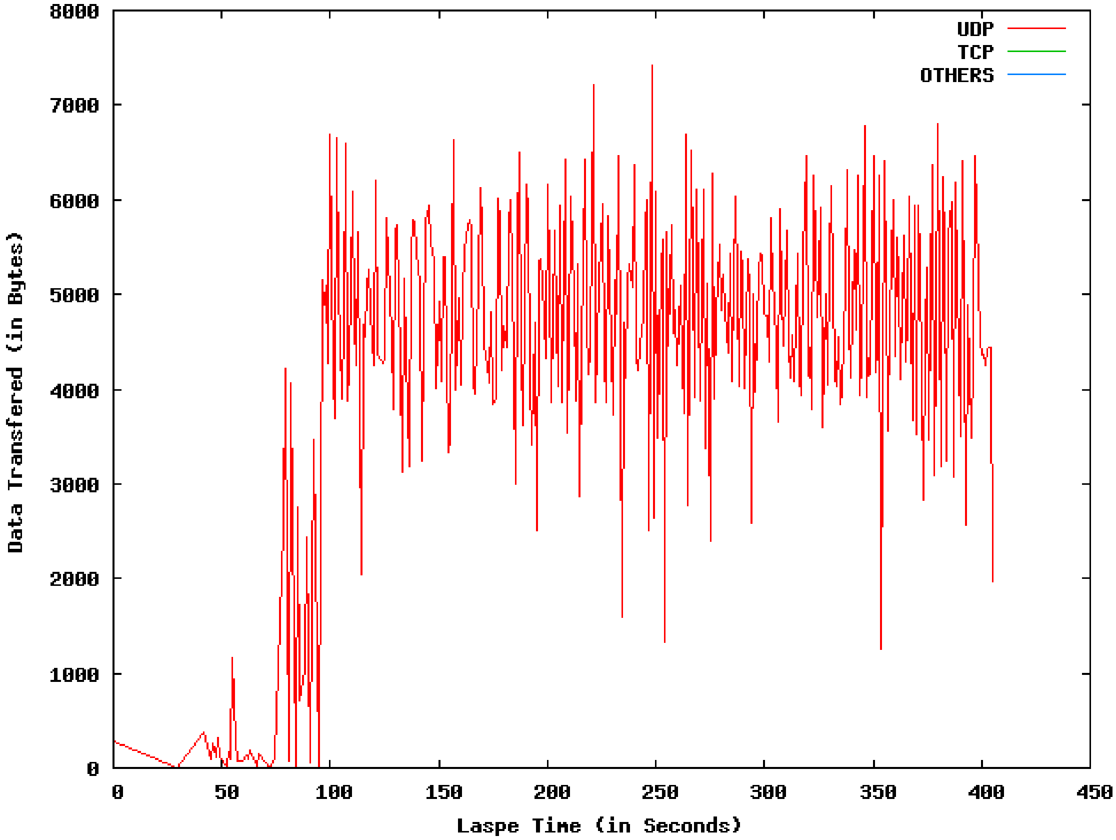


Figure D.19: Battlefield 1942 Traffic Chart

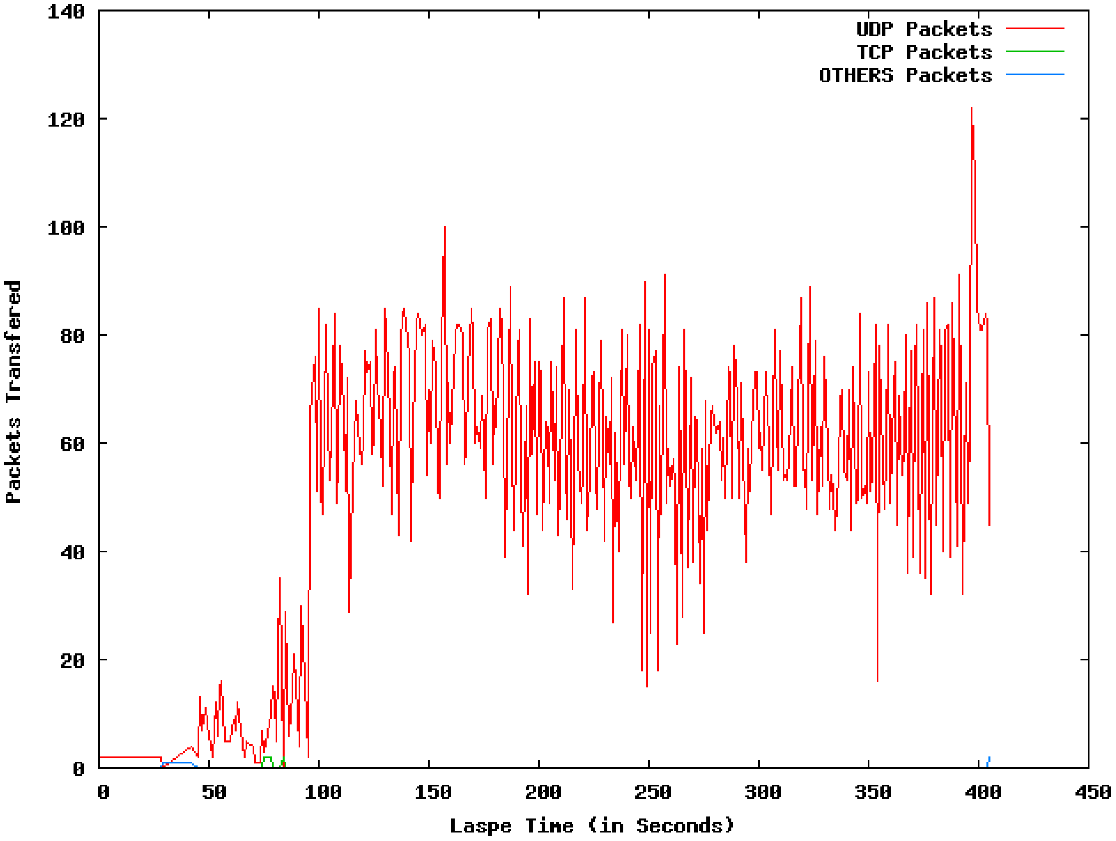


Figure D.20: Battlefield 1942 Packets Chart

D.2.6 Session 6

Game	**Battlefield 1942**
Session	6
Nist.Net Delay Setting	100
Nist.Net Bandwidth Setting	1200
Nist.Net Drop % Setting	1
Nist.Net Duplication % Setting	0
Total Elapsed Time	148
Total Number of Packets	1047
Total Bytes Transfer	71286
Average Number of Packets per Second	7.026845638
Average Bytes Transferred per Second	478.4295302
Average Size per Packet	68.08595989
Total Number of UDP Packets	919
Total Size of UDP Data	68606
Average Number of UDP Packets per Second	6.167785235
Average Size of UDP Transferred per Second	460.442953
Average size per UDP Packet	74.65288357
Total Number of TCP Packets	6
Total Size of TCP Data	240
Average Number of TCP Packets per Second	0.040268456
Average Size of TCP Transferred per Second	1.610738255
Average size per TCP Packet	40
Total Number of OTHERS Packets	122
Total Size of OTHERS Data	2440
Average Number of OTHERS Packets per Second	0.818791946
Average Size of OTHERS Transferred per Second	16.37583893
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

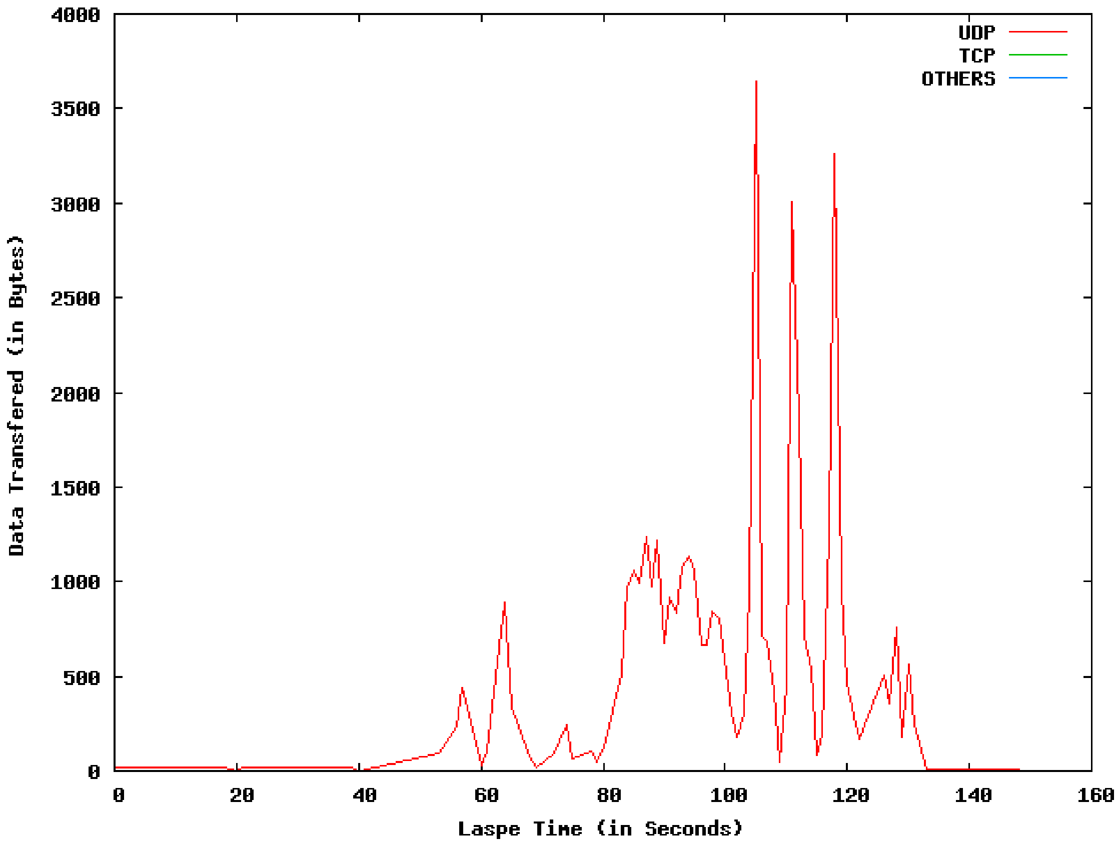


Figure D.21: Battlefield 1942 Traffic Chart

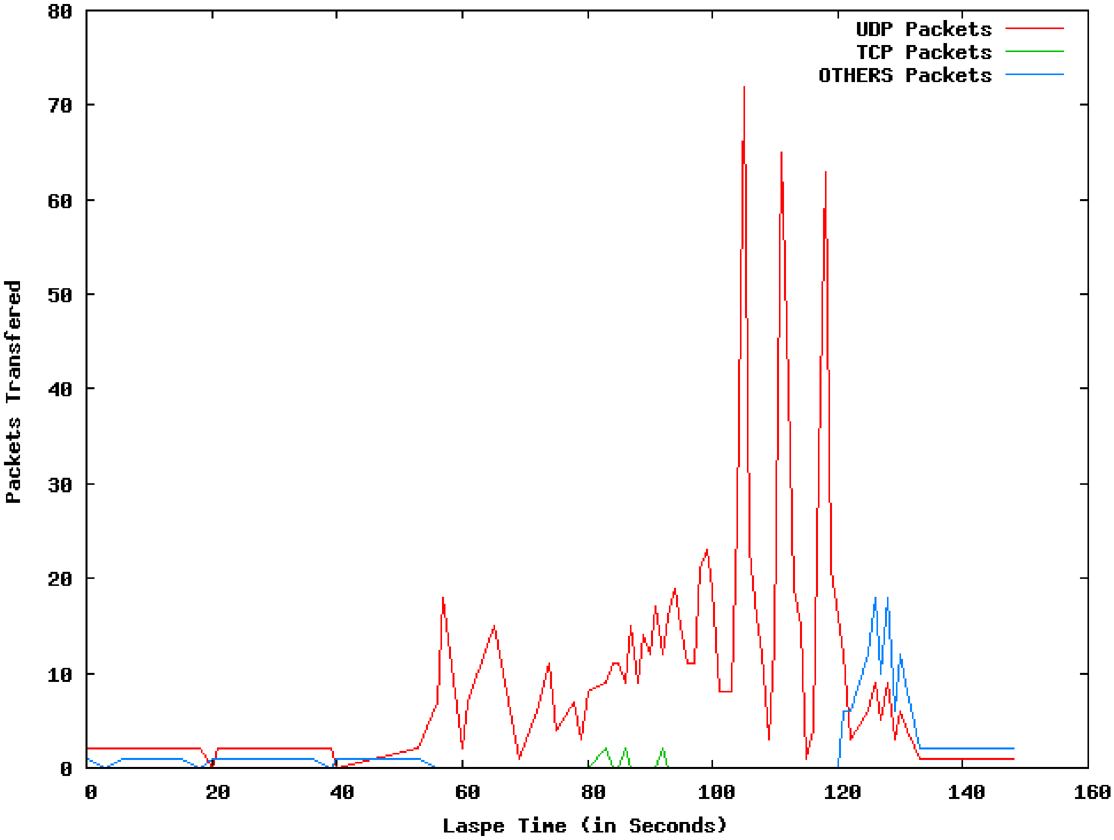


Figure D.22: Battlefield 1942 Packets Chart

D.2.7 Session 8

Game	**Battlefield 1942**
Session	8
Nist.Net Delay Setting	300
Nist.Net Bandwidth Setting	500
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	3
Total Elapsed Time	327
Total Number of Packets	5792
Total Bytes Transfer	550258
Average Number of Packets per Second	17.65853659
Average Bytes Transferred per Second	1677.615854
Average Size per Packet	95.00310773
Total Number of UDP Packets	5692
Total Size of UDP Data	547136
Average Number of UDP Packets per Second	17.35365854
Average Size of UDP Transferred per Second	1668.097561
Average size per UDP Packet	96.12368236
Total Number of TCP Packets	24
Total Size of TCP Data	1602
Average Number of TCP Packets per Second	0.073170732
Average Size of TCP Transferred per Second	4.884146341
Average size per TCP Packet	66.75
Total Number of OTHERS Packets	76
Total Size of OTHERS Data	1520
Average Number of OTHERS Packets per Second	0.231707317
Average Size of OTHERS Transferred per Second	4.634146341
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

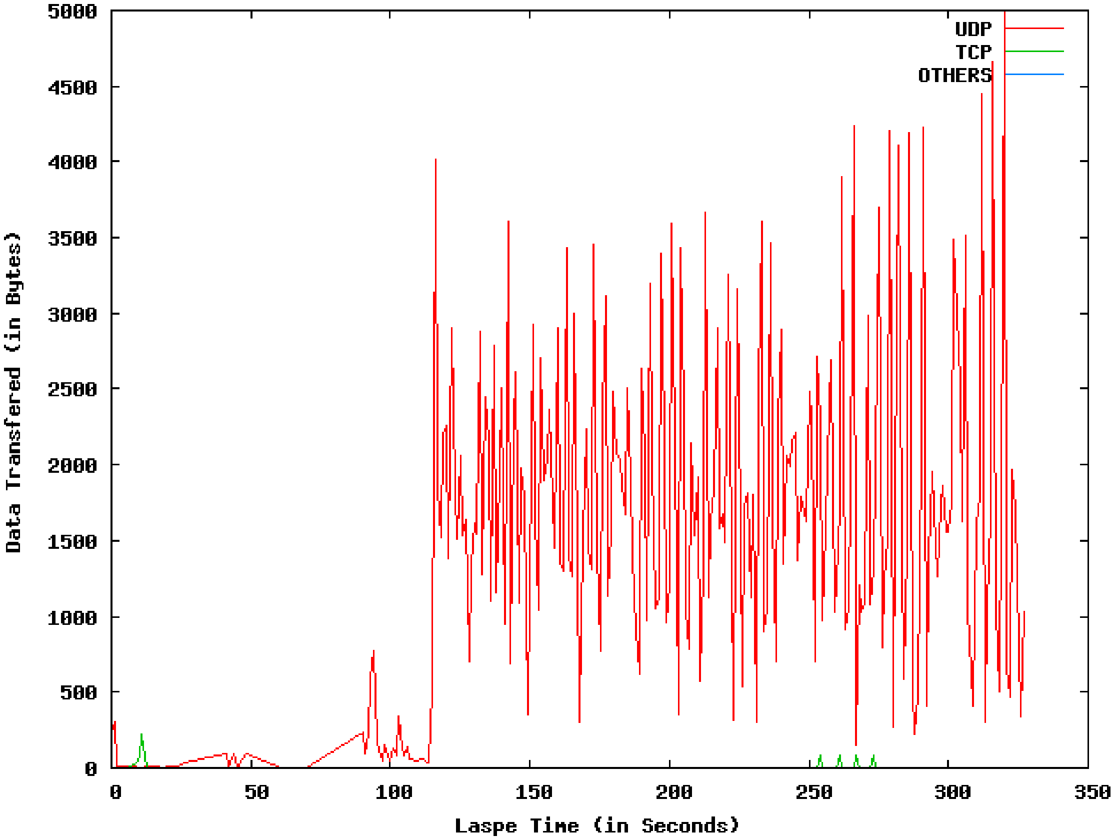


Figure D.23: Battlefield 1942 Traffic Chart

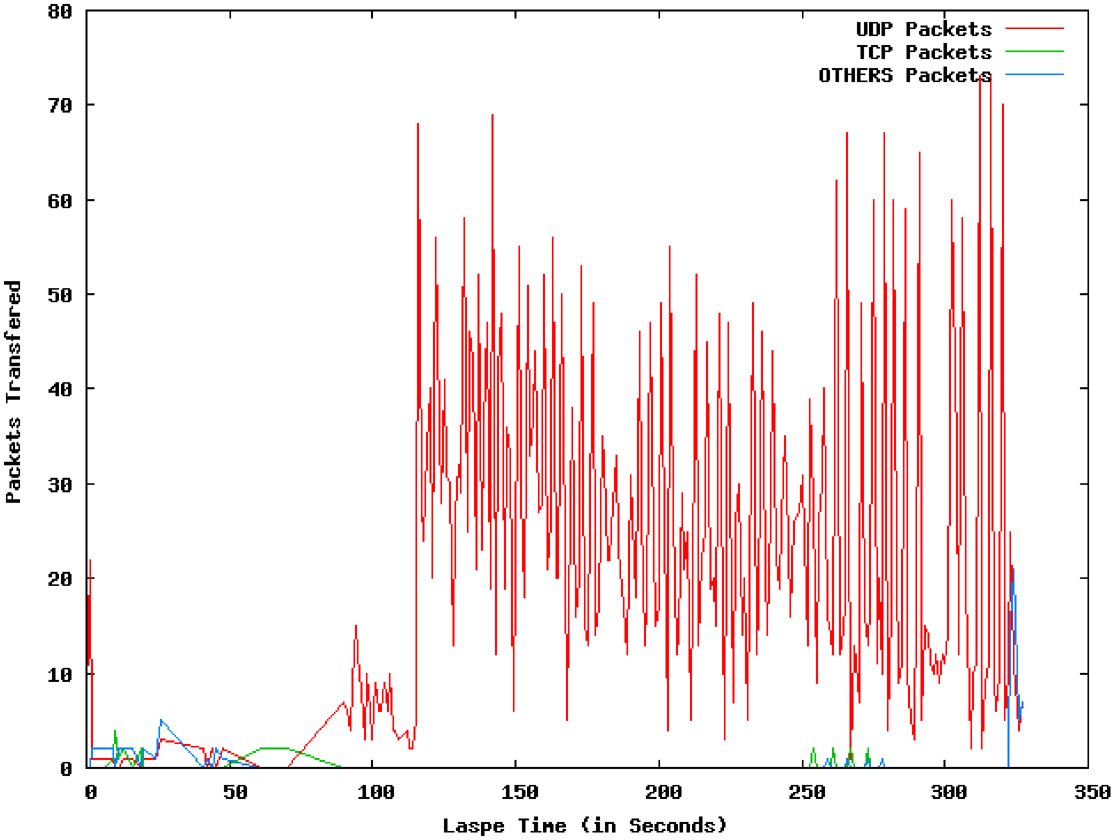


Figure D.24: Battlefield 1942 Packets Chart

D.2.8 Session 9

Game	**Battlefield 1942**
Session	9
Nist.Net Delay Setting	300
Nist.Net Bandwidth Setting	800
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	0
Total Elapsed Time	490
Total Number of Packets	22159
Total Bytes Transfer	2158471
Average Number of Packets per Second	45.13034623
Average Bytes Transferred per Second	4396.071283
Average Size per Packet	97.40832168
Total Number of UDP Packets	22137
Total Size of UDP Data	2157577
Average Number of UDP Packets per Second	45.08553971
Average Size of UDP Transferred per Second	4394.250509
Average size per UDP Packet	97.46474229
Total Number of TCP Packets	11
Total Size of TCP Data	674
Average Number of TCP Packets per Second	0.022403259
Average Size of TCP Transferred per Second	1.372708758
Average size per TCP Packet	61.27272727
Total Number of OTHERS Packets	11
Total Size of OTHERS Data	220
Average Number of OTHERS Packets per Second	0.022403259
Average Size of OTHERS Transferred per Second	0.448065173
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

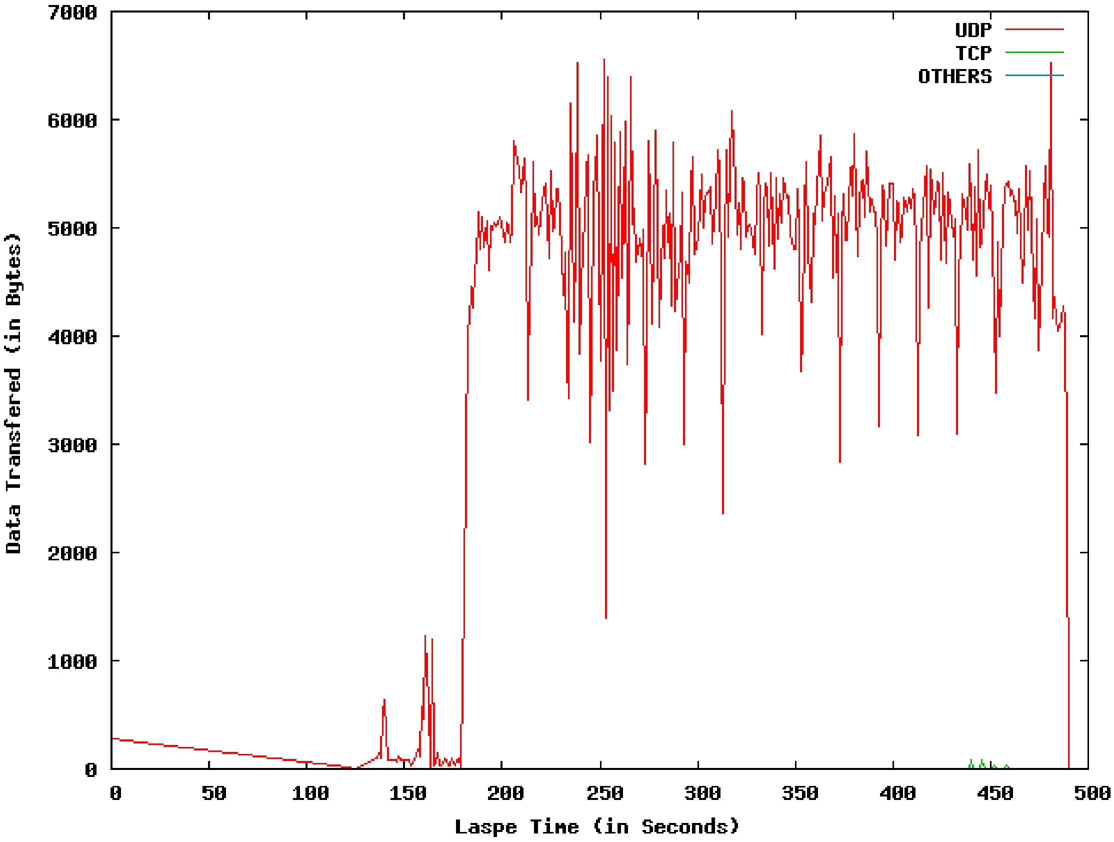


Figure D.25: Battlefield 1942 Traffic Chart

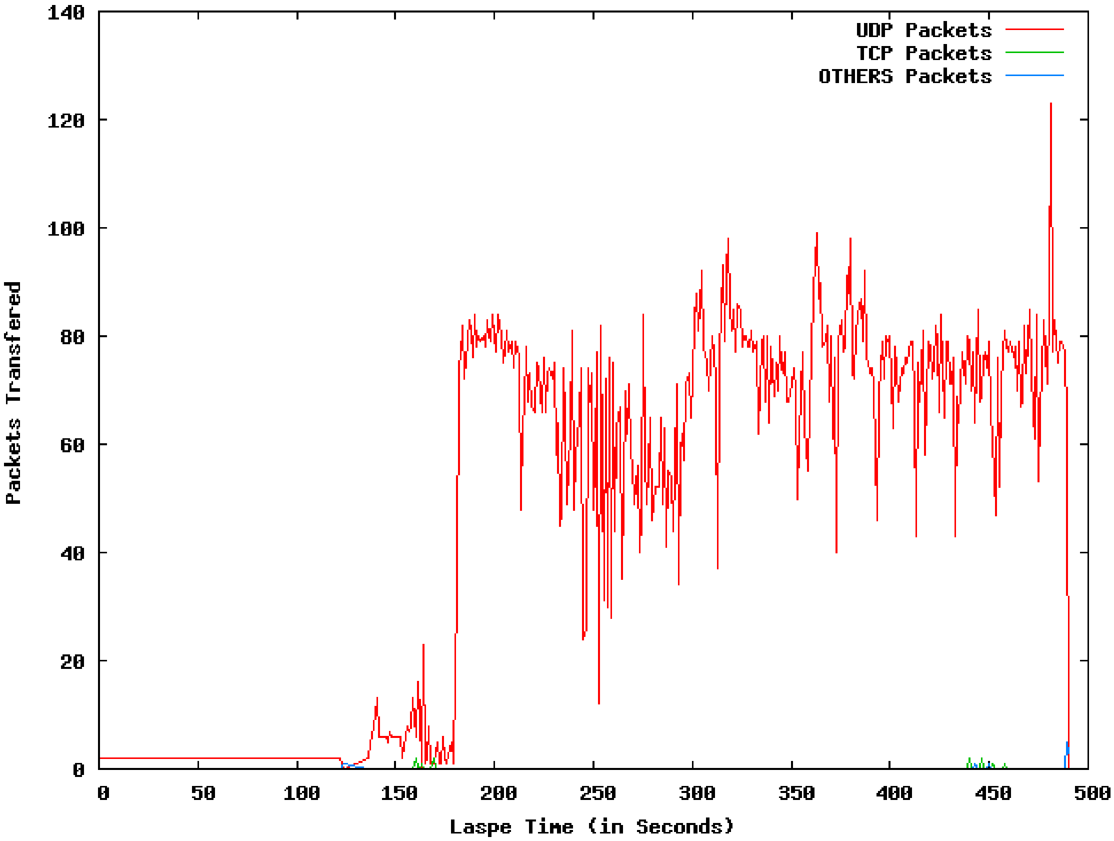


Figure D.26: Battlefield 1942 Packets Chart

D.2.9 Session 10

Game	**Battlefield 1942**
Session	10
Nist.Net Delay Setting	200
Nist.Net Bandwidth Setting	5000
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	0
Total Elapsed Time	447
Total Number of Packets	23301
Total Bytes Transfer	2275889
Average Number of Packets per Second	52.01116071
Average Bytes Transferred per Second	5080.109375
Average Size per Packet	97.67344749
Total Number of UDP Packets	23286
Total Size of UDP Data	2275489
Average Number of UDP Packets per Second	51.97767857
Average Size of UDP Transferred per Second	5079.216518
Average size per UDP Packet	97.71918749
Total Number of TCP Packets	5
Total Size of TCP Data	200
Average Number of TCP Packets per Second	0.011160714
Average Size of TCP Transferred per Second	0.446428571
Average size per TCP Packet	40
Total Number of OTHERS Packets	10
Total Size of OTHERS Data	200
Average Number of OTHERS Packets per Second	0.022321429
Average Size of OTHERS Transferred per Second	0.446428571
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

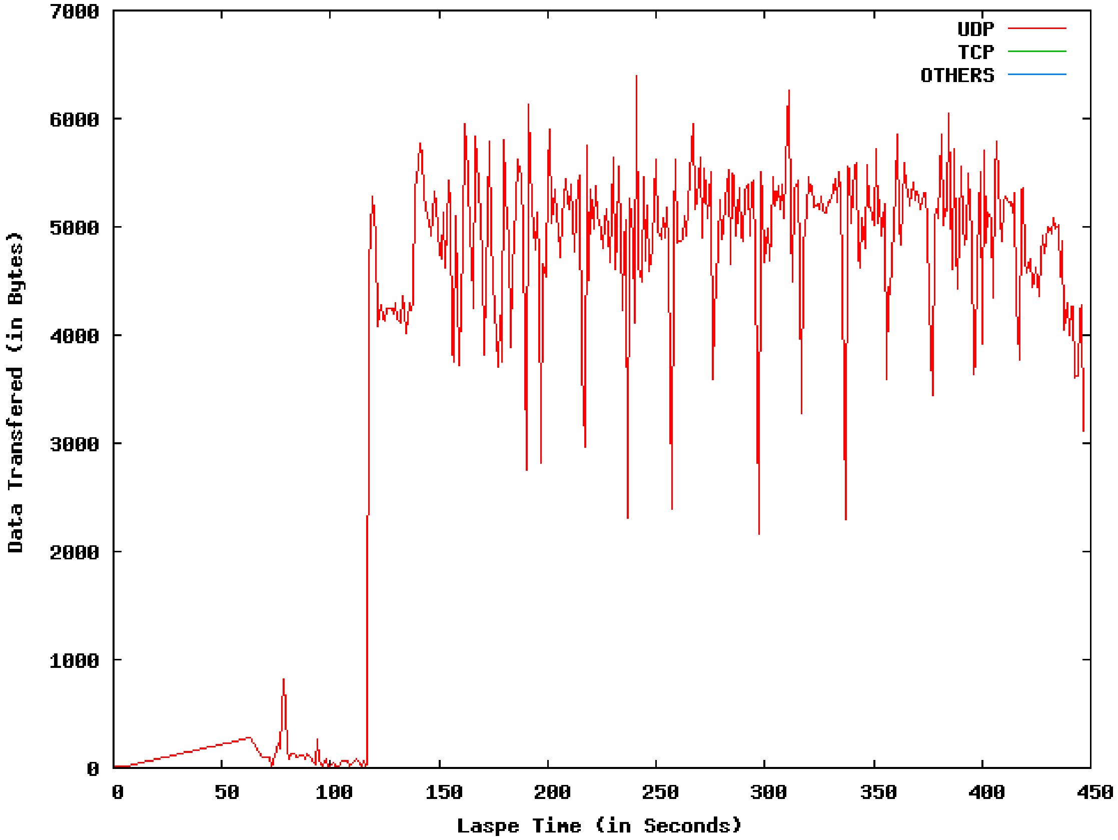


Figure D.27: Battlefield 1942 Traffic Chart

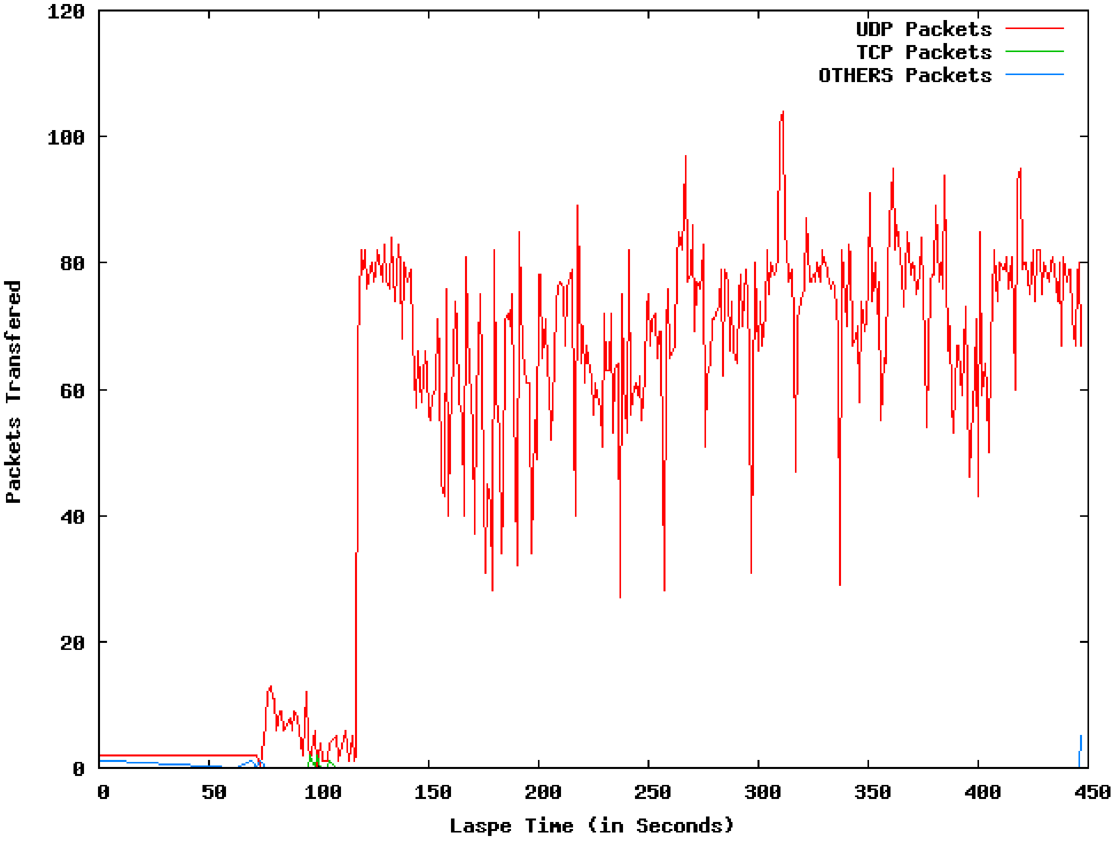


Figure D.28: Battlefield 1942 Packets Chart

D.2.10 Session 11

Game	**Battlefield 1942**
Session	11
Nist.Net Delay Setting	100
Nist.Net Bandwidth Setting	5000
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	0
Total Elapsed Time	372
Total Number of Packets	21939
Total Bytes Transfer	2130142
Average Number of Packets per Second	58.81769437
Average Bytes Transferred per Second	5710.836461
Average Size per Packet	97.09385113
Total Number of UDP Packets	21907
Total Size of UDP Data	2128785
Average Number of UDP Packets per Second	58.73190349
Average Size of UDP Transferred per Second	5707.198391
Average size per UDP Packet	97.17373442
Total Number of TCP Packets	16
Total Size of TCP Data	1037
Average Number of TCP Packets per Second	0.042895442
Average Size of TCP Transferred per Second	2.780160858
Average size per TCP Packet	64.8125
Total Number of OTHERS Packets	16
Total Size of OTHERS Data	320
Average Number of OTHERS Packets per Second	0.042895442
Average Size of OTHERS Transferred per Second	0.857908847
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

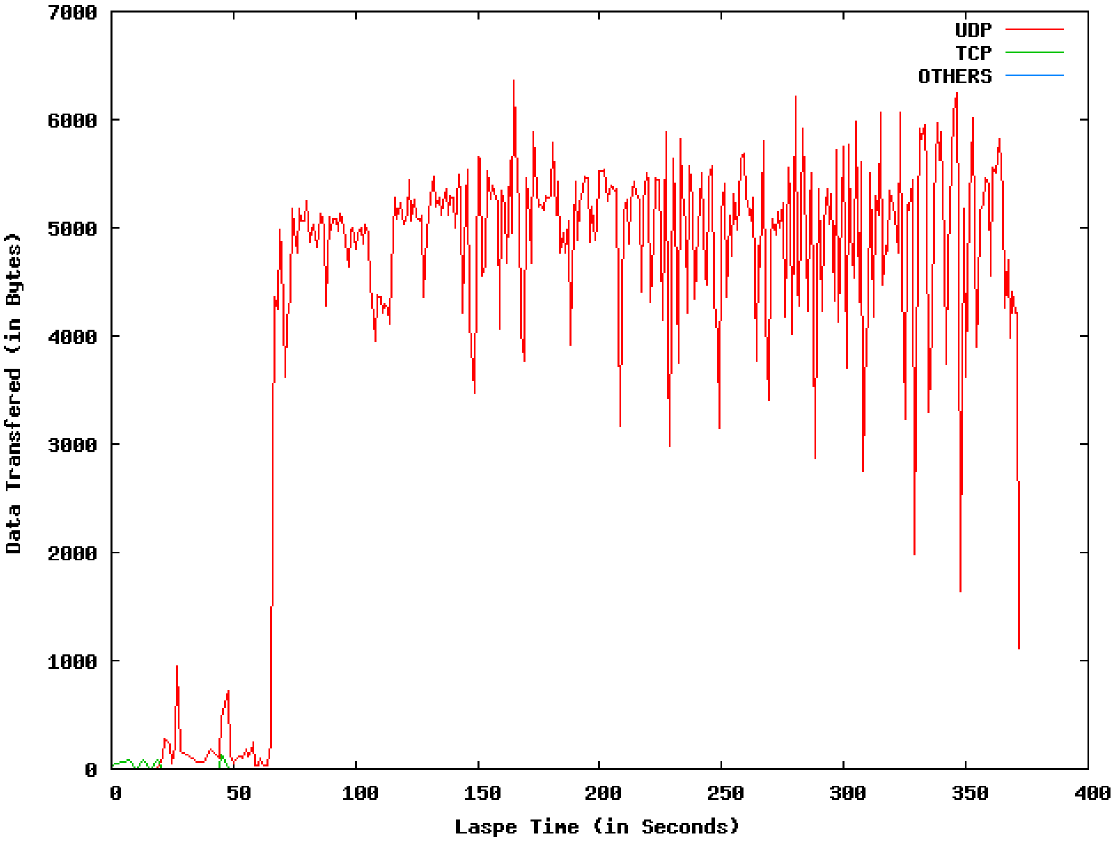


Figure D.29: Battlefield 1942 Traffic Chart

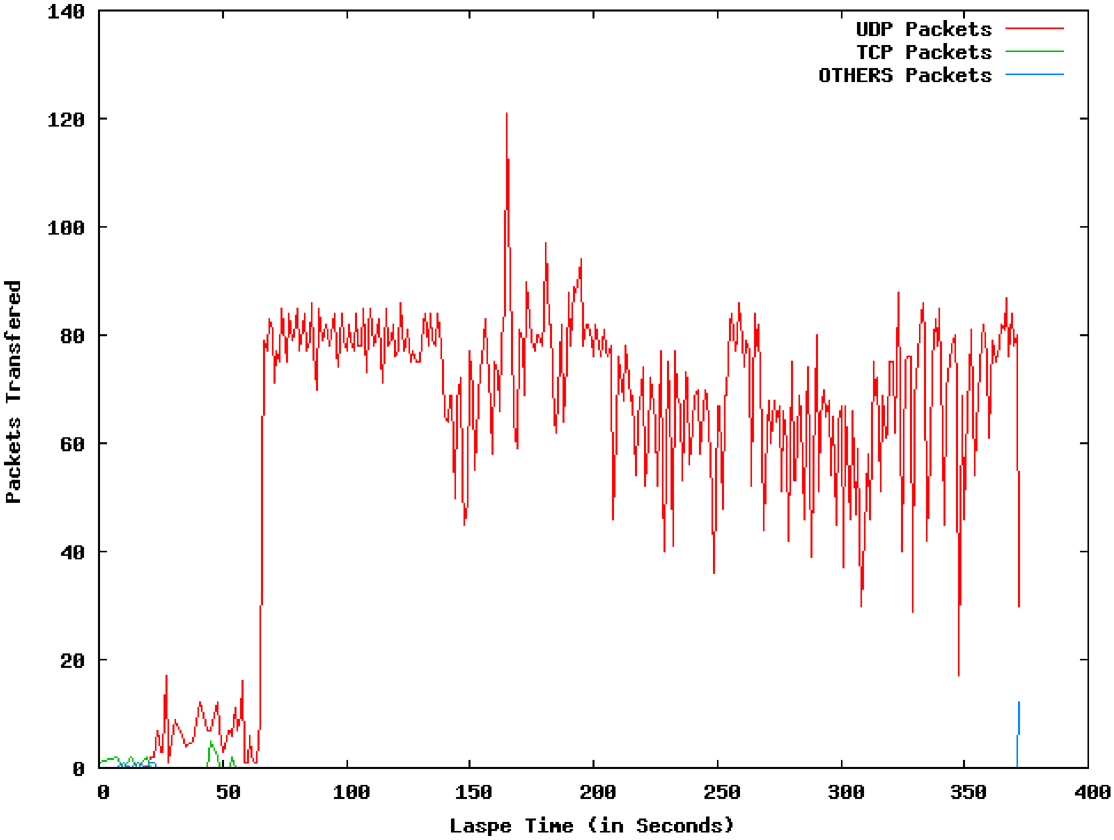


Figure D.30: Battlefield 1942 Packets Chart

D.2.11 Session 12

Game	**Battlefield 1942**
Session	12
Nist.Net Delay Setting	100
Nist.Net Bandwidth Setting	5000
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	0
Total Elapsed Time	436
Total Number of Packets	24599
Total Bytes Transfer	2517614
Average Number of Packets per Second	56.29061785
Average Bytes Transferred per Second	5761.130435
Average Size per Packet	102.3461929
Total Number of UDP Packets	24583
Total Size of UDP Data	2516761
Average Number of UDP Packets per Second	56.25400458
Average Size of UDP Transferred per Second	5759.17849
Average size per UDP Packet	102.3781068
Total Number of TCP Packets	13
Total Size of TCP Data	793
Average Number of TCP Packets per Second	0.029748284
Average Size of TCP Transferred per Second	1.814645309
Average size per TCP Packet	61
Total Number of OTHERS Packets	3
Total Size of OTHERS Data	60
Average Number of OTHERS Packets per Second	0.006864989
Average Size of OTHERS Transferred per Second	0.137299771
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

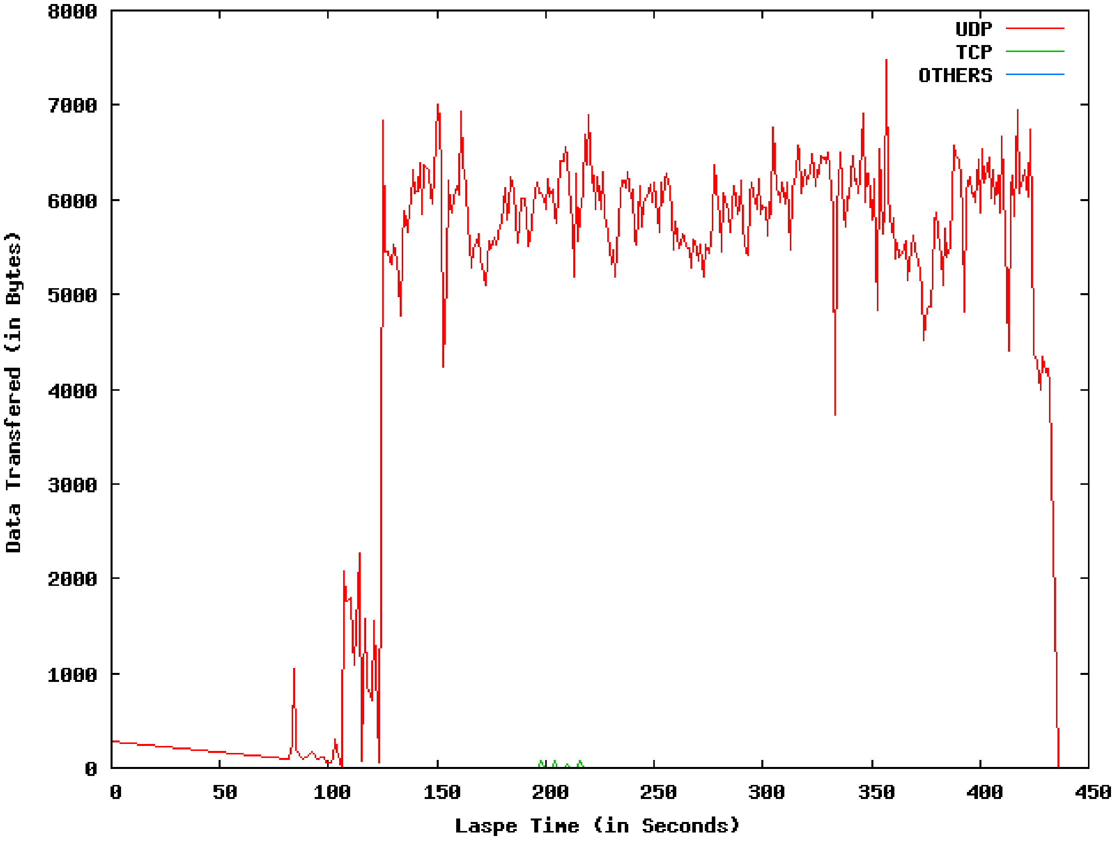


Figure D.31: Battlefield 1942 Traffic Chart

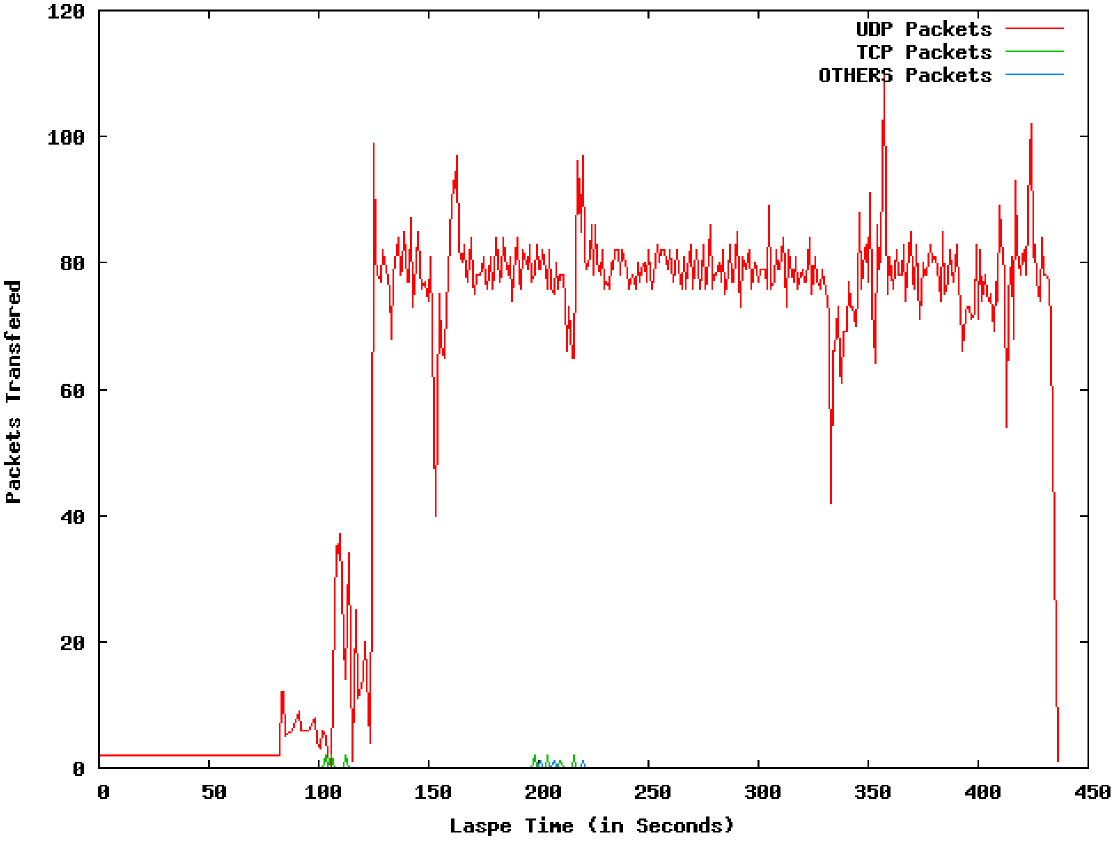


Figure D.32: Battlefield 1942 Packets Chart

D.2.12 Session 13

Game	**Battlefield 1942**
Session	13
Nist.Net Delay Setting	100
Nist.Net Bandwidth Setting	6000
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	0
Total Elapsed Time	357
Total Number of Packets	24324
Total Bytes Transfer	2578957
Average Number of Packets per Second	67.94413408
Average Bytes Transferred per Second	7203.790503
Average Size per Packet	106.0252014
Total Number of UDP Packets	24303
Total Size of UDP Data	2577945
Average Number of UDP Packets per Second	67.88547486
Average Size of UDP Transferred per Second	7200.963687
Average size per UDP Packet	106.0751759
Total Number of TCP Packets	14
Total Size of TCP Data	872
Average Number of TCP Packets per Second	0.039106145
Average Size of TCP Transferred per Second	2.43575419
Average size per TCP Packet	62.28571429
Total Number of OTHERS Packets	7
Total Size of OTHERS Data	140
Average Number of OTHERS Packets per Second	0.019553073
Average Size of OTHERS Transferred per Second	0.391061453
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

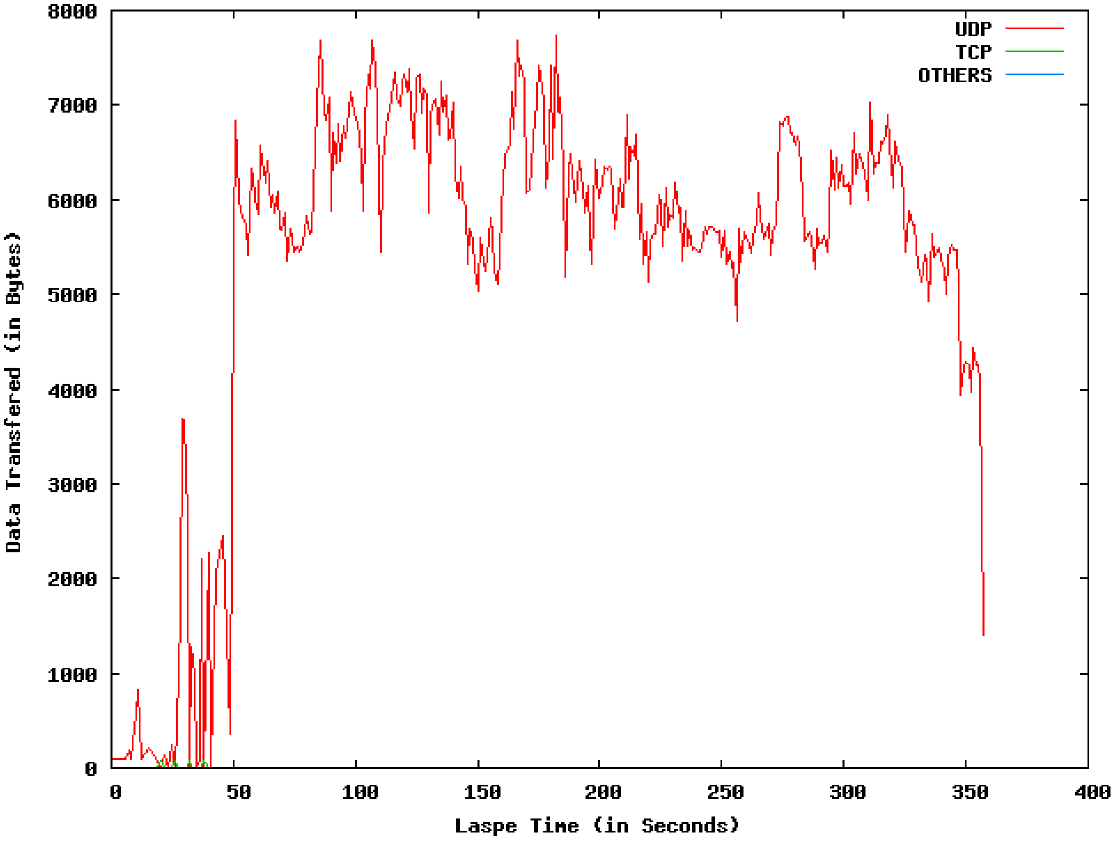


Figure D.33: Battlefield 1942 Traffic Chart

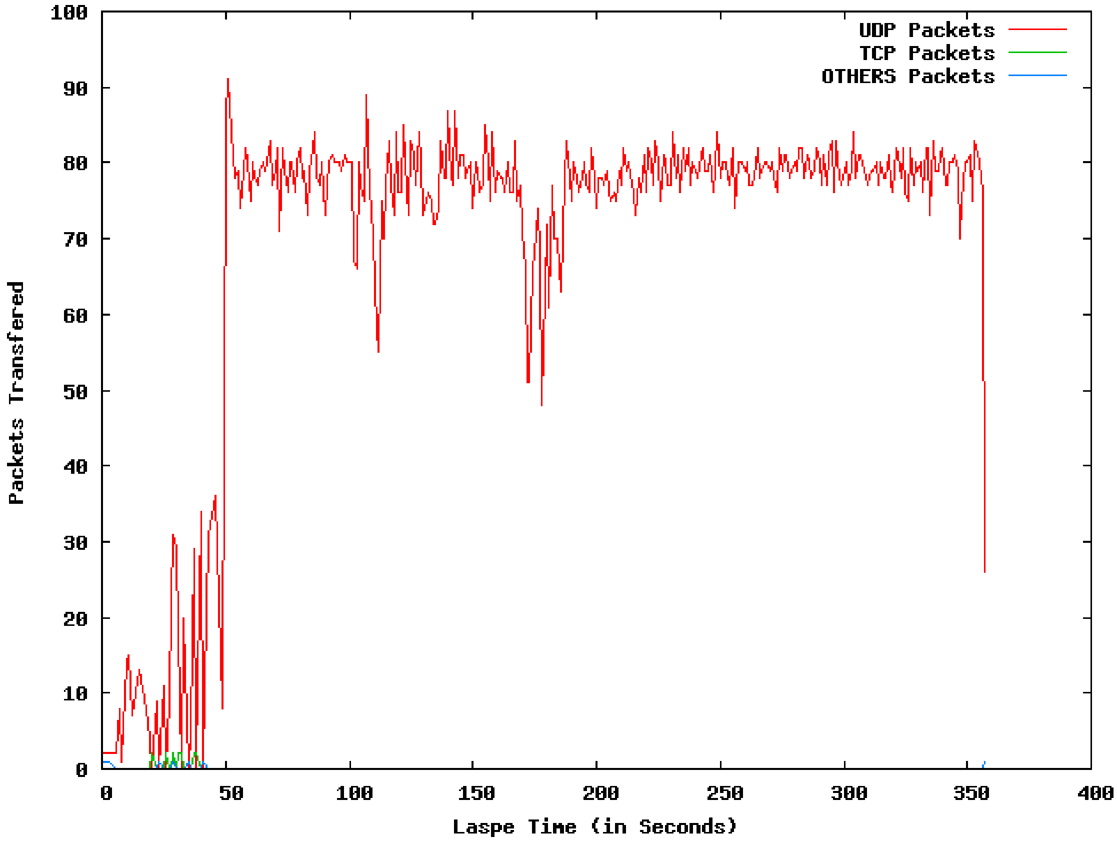


Figure D.34: Battlefield 1942 Packets Chart

D.2.13 Session 14

Game	**Battlefield 1942**
Session	14
Nist.Net Delay Setting	200
Nist.Net Bandwidth Setting	7000
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	0
Total Elapsed Time	492
Total Number of Packets	24805
Total Bytes Transfer	2798819
Average Number of Packets per Second	50.00
Average Bytes Transferred per Second	5677.00
Average Size per Packet	112.00
Total Number of UDP Packets	24759
Total Size of UDP Data	2797207
Average Number of UDP Packets per Second	50.00
Average Size of UDP Transferred per Second	5673.00
Average size per UDP Packet	112.00
Total Number of TCP Packets	19
Total Size of TCP Data	1072
Average Number of TCP Packets per Second	0.00
Average Size of TCP Transferred per Second	2.00
Average size per TCP Packet	56.00
Total Number of OTHERS Packets	27
Total Size of OTHERS Data	540
Average Number of OTHERS Packets per Second	0.00
Average Size of OTHERS Transferred per Second	1.00
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

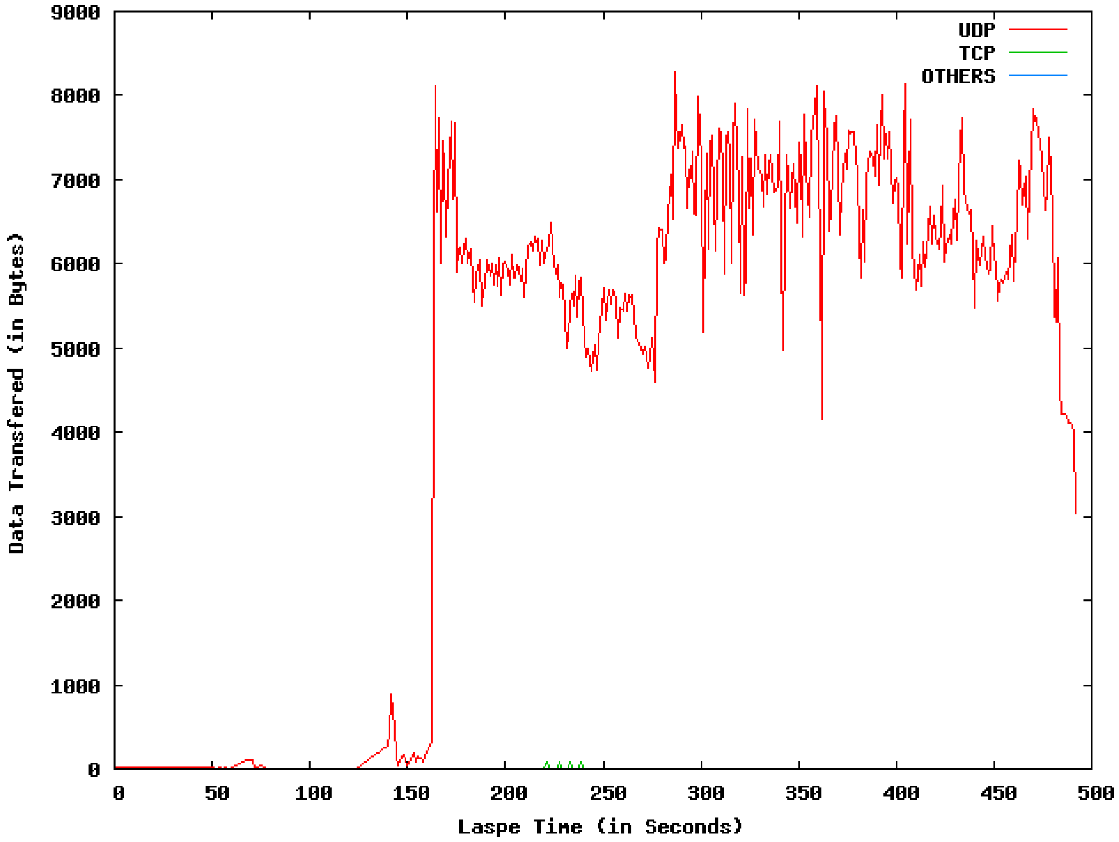


Figure D.35: Battlefield 1942 Traffic Chart

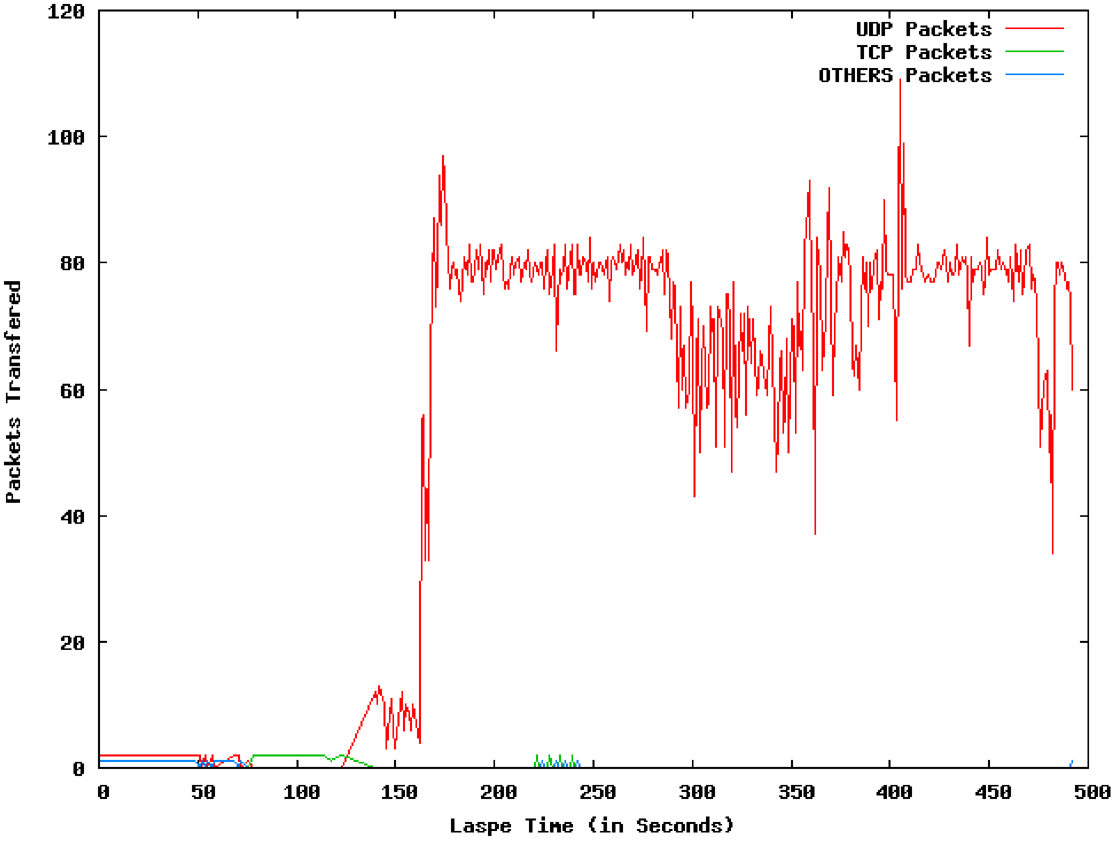


Figure D.36: Battlefield 1942 Packets Chart

D.3 Mechwarrior:4 Mercenaries

D.3.1 Session 1

Game	Mechwarrior:4 Mercenaries
Session	1
Nist.Net Delay Setting	0
Nist.Net Bandwidth Setting	12500000
Nist.Net Drop % Setting	0
Nist.Net Duplication % Setting	0
Total Elapsed Time	1517
Total Number of Packets	12498
Total Bytes Transfer	1935363
Average Number of Packets per Second	8.233201581
Average Bytes Transferred per Second	1274.942688
Average Size per Packet	154.8538166
Total Number of UDP Packets	10544
Total Size of UDP Data	1638338
Average Number of UDP Packets per Second	6.945981555
Average Size of UDP Transferred per Second	1079.274045
Average size per UDP Packet	155.3810698
Total Number of TCP Packets	1895
Total Size of TCP Data	292235
Average Number of TCP Packets per Second	1.248353096
Average Size of TCP Transferred per Second	192.5131752
Average size per TCP Packet	154.2137203
Total Number of OTHERS Packets	59
Total Size of OTHERS Data	4790
Average Number of OTHERS Packets per Second	0.03886693
Average Size of OTHERS Transferred per Second	3.155467721
Average size per OTHERS Packet	81.18644068

Traffic Graph

Packets Graph

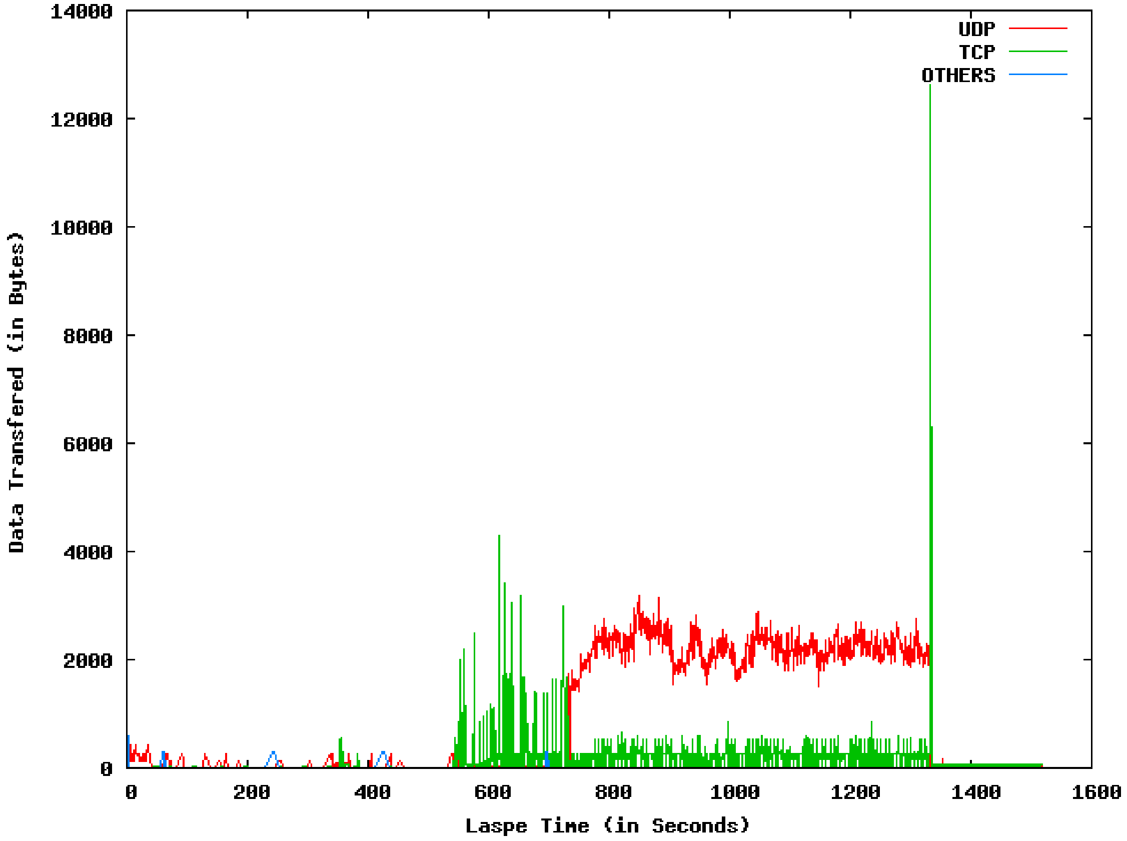


Figure D.37: Mechwarrior: 4 Mercenaries Traffic Chart

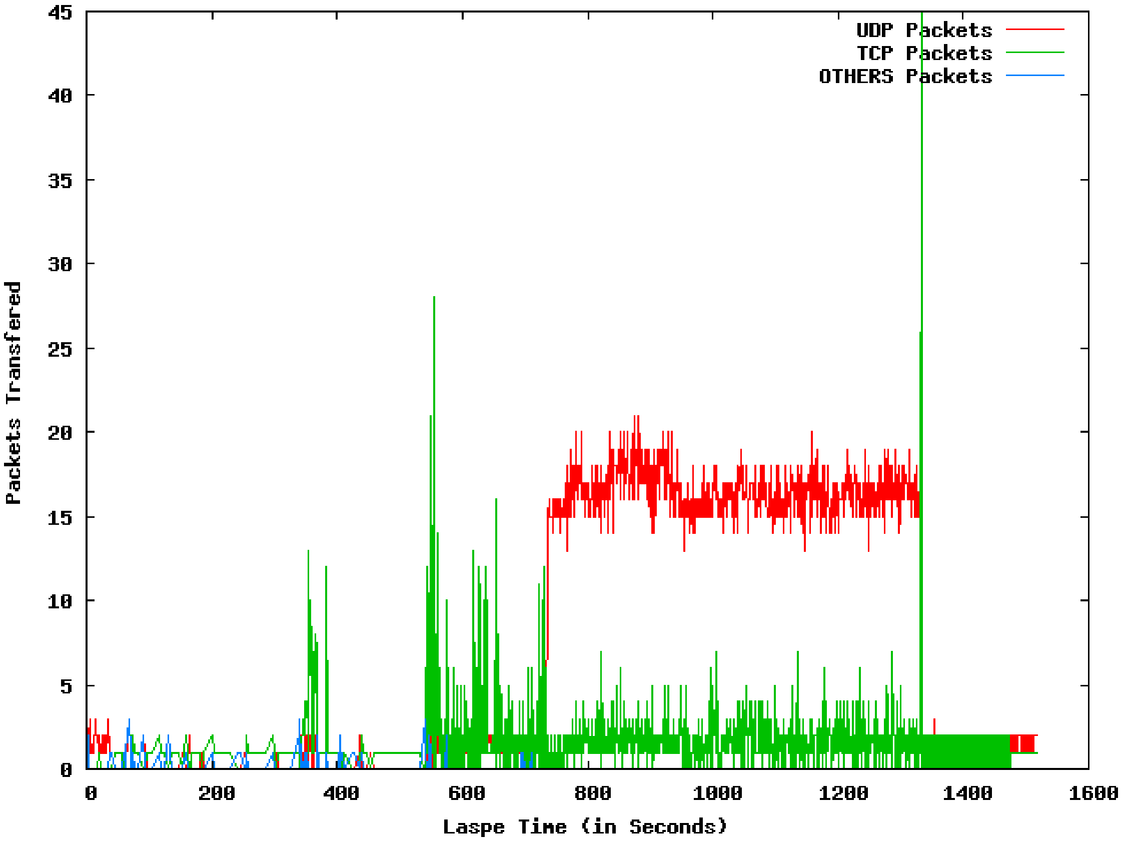


Figure D.38: Mechwarrior: 4 Mercenaries Packets Chart

D.3.2 Session 3

Game	Mechwarrior:4 Mercenaries
Session	3
Nist.Net Delay Setting	500
Nist.Net Bandwidth Setting	1200
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	0
Total Elapsed Time	1173
Total Number of Packets	11088
Total Bytes Transfer	1334912
Average Number of Packets per Second	9.00
Average Bytes Transferred per Second	1137.00
Average Size per Packet	120.00
Total Number of UDP Packets	7759
Total Size of UDP Data	846038
Average Number of UDP Packets per Second	6.00
Average Size of UDP Transferred per Second	720.00
Average size per UDP Packet	109.00
Total Number of TCP Packets	3329
Total Size of TCP Data	488874
Average Number of TCP Packets per Second	2.00
Average Size of TCP Transferred per Second	416.00
Average size per TCP Packet	146.00
Total Number of OTHERS Packets	0
Total Size of OTHERS Data	0
Average Number of OTHERS Packets per Second	0
Average Size of OTHERS Transferred per Second	0
Average size per OTHERS Packet	0

Traffic Graph

Packets Graph

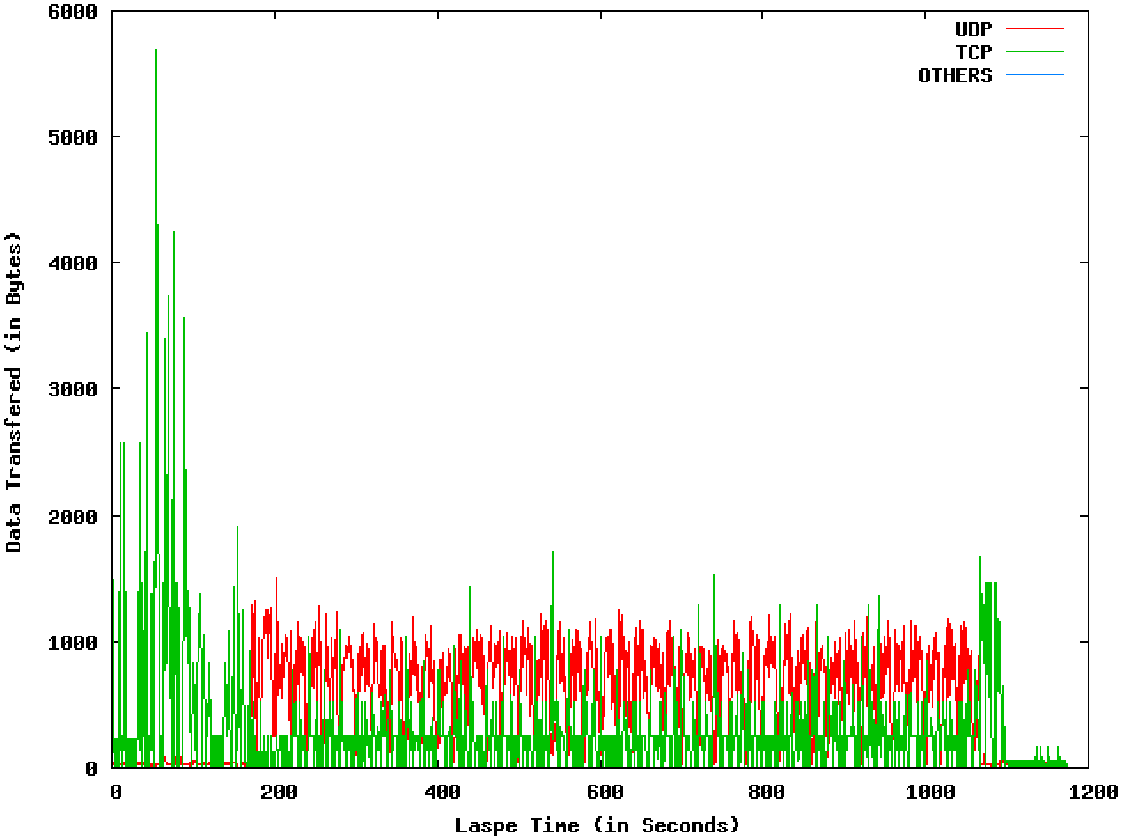


Figure D.39: Mechwarrior: 4 Mercenaries Traffic Chart

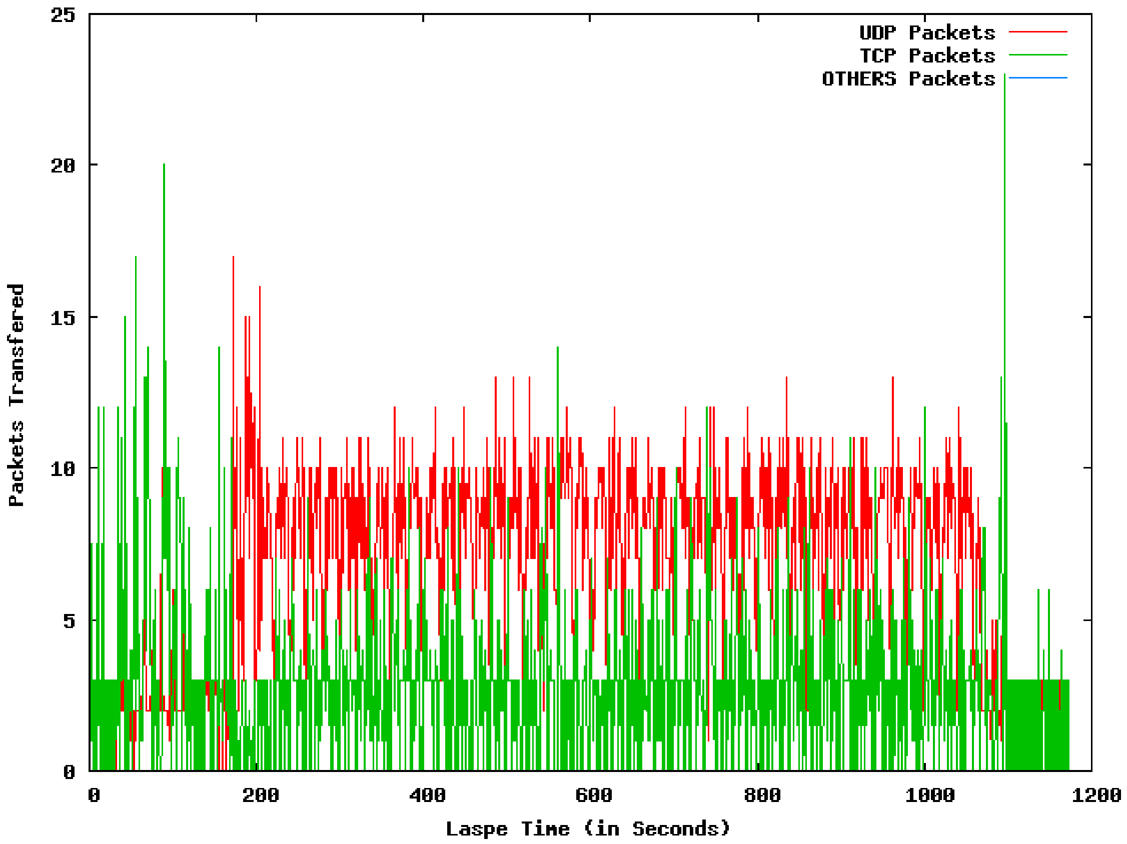


Figure D.40: Mechwarrior: 4 Mercenaries Packets Chart

D.3.3 Session 4

Game	Mechwarrior:4 Mercenaries
Session	4
Nist.Net Delay Setting	400
Nist.Net Bandwidth Setting	800
Nist.Net Drop % Setting	10
Nist.Net Duplication % Setting	3
Total Elapsed Time	1705
Total Number of Packets	10185
Total Bytes Transfer	1098165
Average Number of Packets per Second	5.97010551
Average Bytes Transferred per Second	643.7075029
Average Size per Packet	107.8217968
Total Number of UDP Packets	5831
Total Size of UDP Data	400273
Average Number of UDP Packets per Second	3.417936694
Average Size of UDP Transferred per Second	234.626612
Average size per UDP Packet	68.64568685
Total Number of TCP Packets	4236
Total Size of TCP Data	695532
Average Number of TCP Packets per Second	2.483001172
Average Size of TCP Transferred per Second	407.6975381
Average size per TCP Packet	164.1954674
Total Number of OTHERS Packets	118
Total Size of OTHERS Data	2360
Average Number of OTHERS Packets per Second	0.069167644
Average Size of OTHERS Transferred per Second	1.383352872
Average size per OTHERS Packet	20

Traffic Graph

Packets Graph

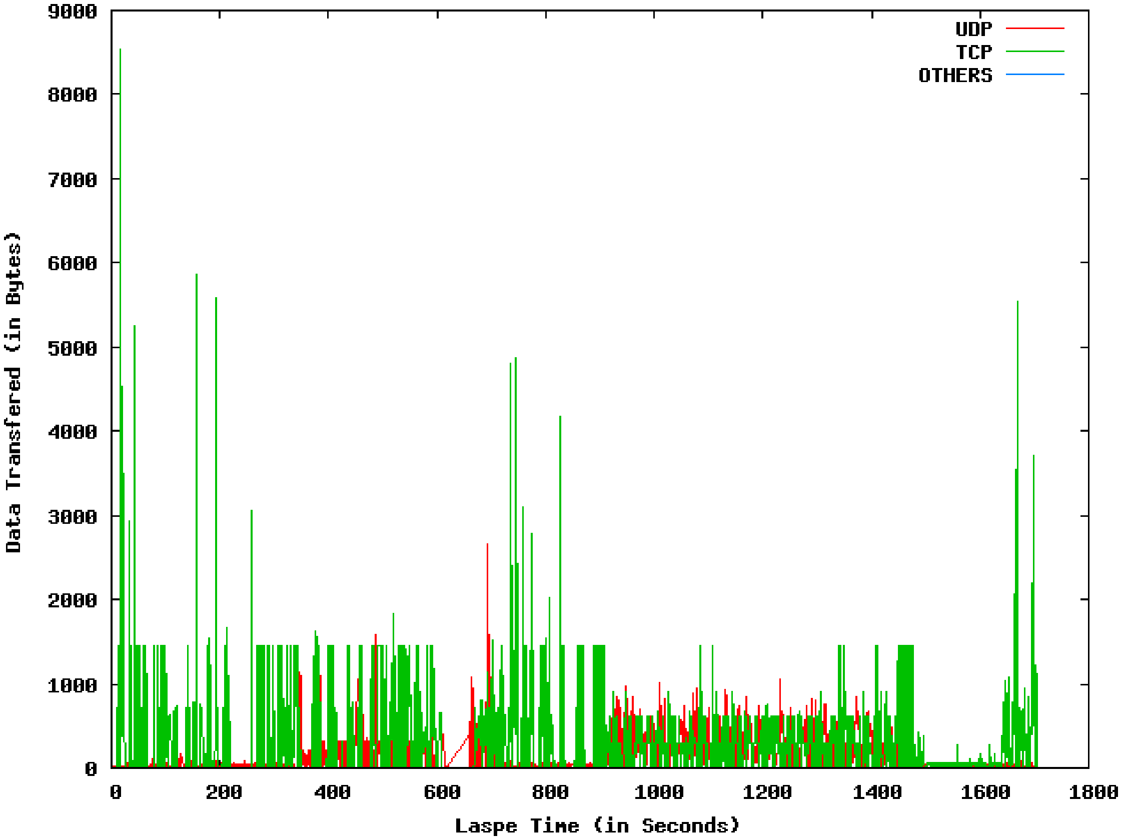


Figure D.41: Mechwarrior: 4 Mercenaries Traffic Chart

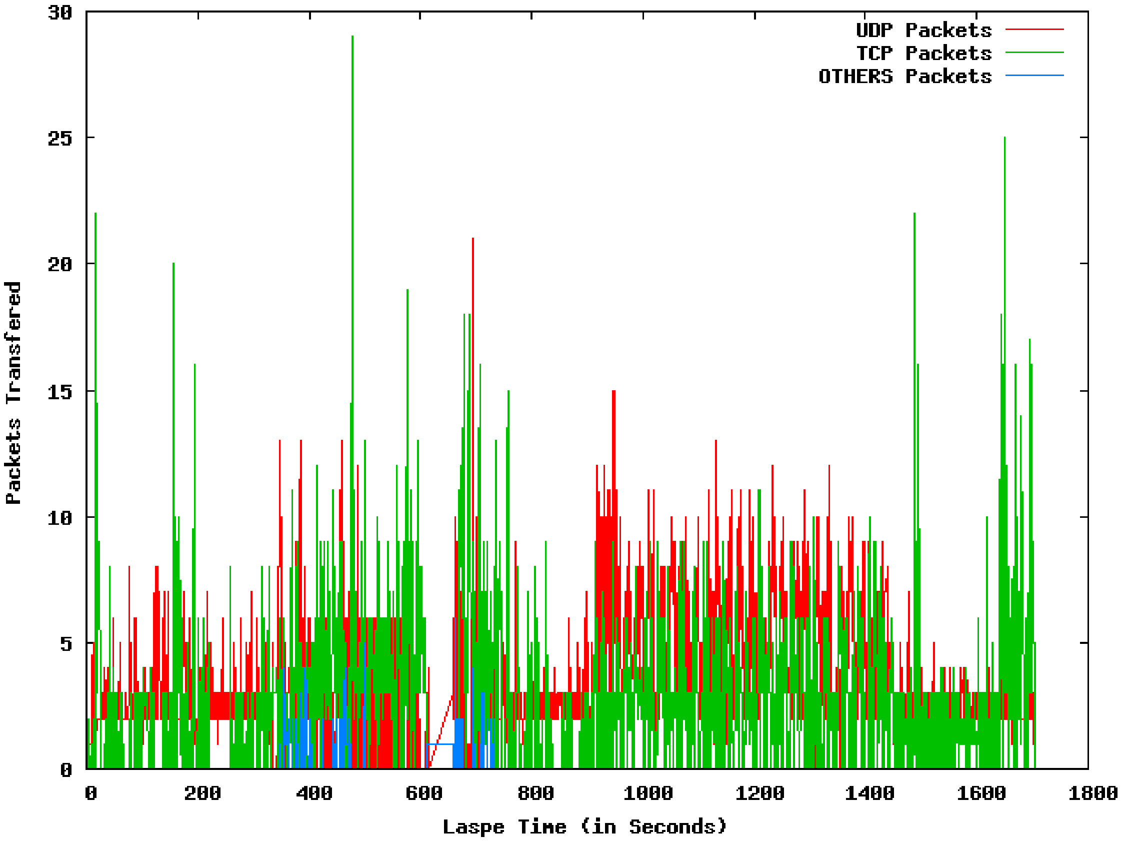


Figure D.42: Mechwarrior: 4 Mercenaries Packets Chart

D.3.4 Session 5

Game	Mechwarrior:4 Mercenaries
Session	5
Nist.Net Delay Setting	300
Nist.Net Bandwidth Setting	800
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	3
Total Elapsed Time	696
Total Number of Packets	5736
Total Bytes Transfer	555825
Average Number of Packets per Second	8.229555237
Average Bytes Transferred per Second	797.4533716
Average Size per Packet	96.90115063
Total Number of UDP Packets	3622
Total Size of UDP Data	296572
Average Number of UDP Packets per Second	5.196556671
Average Size of UDP Transferred per Second	425.4978479
Average size per UDP Packet	81.88072888
Total Number of TCP Packets	2114
Total Size of TCP Data	259253
Average Number of TCP Packets per Second	3.032998565
Average Size of TCP Transferred per Second	371.9555237
Average size per TCP Packet	122.6362346
Total Number of OTHERS Packets	0
Total Size of OTHERS Data	0
Average Number of OTHERS Packets per Second	0
Average Size of OTHERS Transferred per Second	0
Average size per OTHERS Packet	0

Traffic Graph

Packets Graph

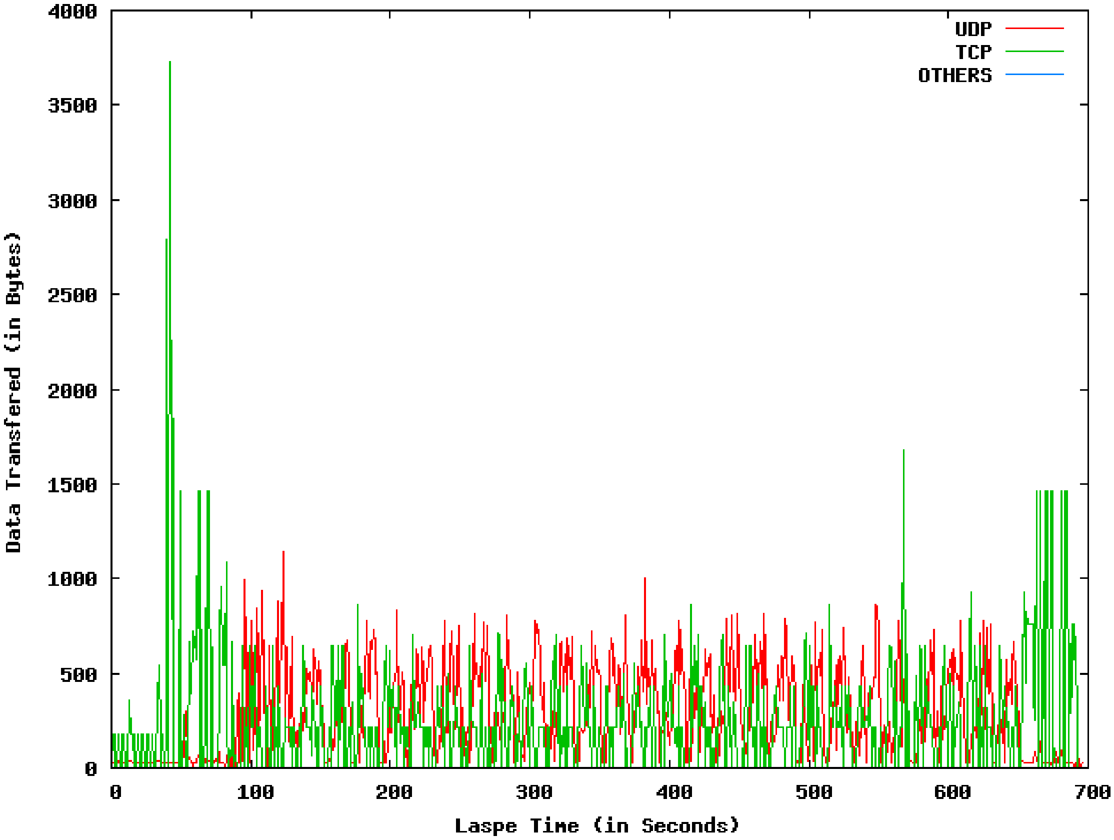


Figure D.43: Mechwarrior: 4 Mercenaries Traffic Chart

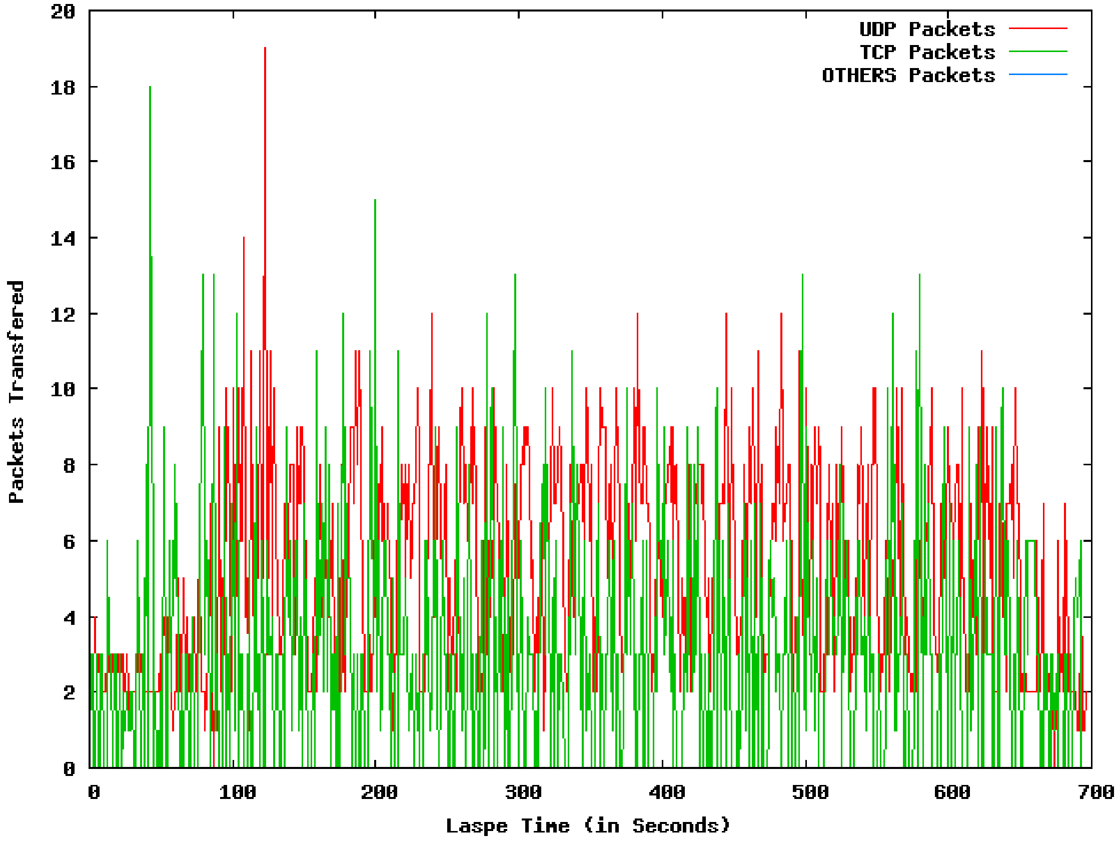


Figure D.44: Mechwarrior: 4 Mercenaries Packets Chart

D.3.5 Session 6

Game	Mechwarrior:4 Mercenaries
Session	6
Nist.Net Delay Setting	200
Nist.Net Bandwidth Setting	2000
Nist.Net Drop % Setting	5
Nist.Net Duplication % Setting	3
Total Elapsed Time	86396
Total Number of Packets	11423
Total Bytes Transfer	1168985
Average Number of Packets per Second	0.132215239
Average Bytes Transferred per Second	13.53038879
Average Size per Packet	102.3360763
Total Number of UDP Packets	7345
Total Size of UDP Data	751200
Average Number of UDP Packets per Second	0.085014526
Average Size of UDP Transferred per Second	8.694746345
Average size per UDP Packet	102.2736555
Total Number of TCP Packets	3979
Total Size of TCP Data	398405
Average Number of TCP Packets per Second	0.04605484
Average Size of TCP Transferred per Second	4.611329097
Average size per TCP Packet	100.1269163
Total Number of OTHERS Packets	99
Total Size of OTHERS Data	19380
Average Number of OTHERS Packets per Second	0.001145873
Average Size of OTHERS Transferred per Second	0.224313344
Average size per OTHERS Packet	195.7575758

Traffic Graph

Packets Graph

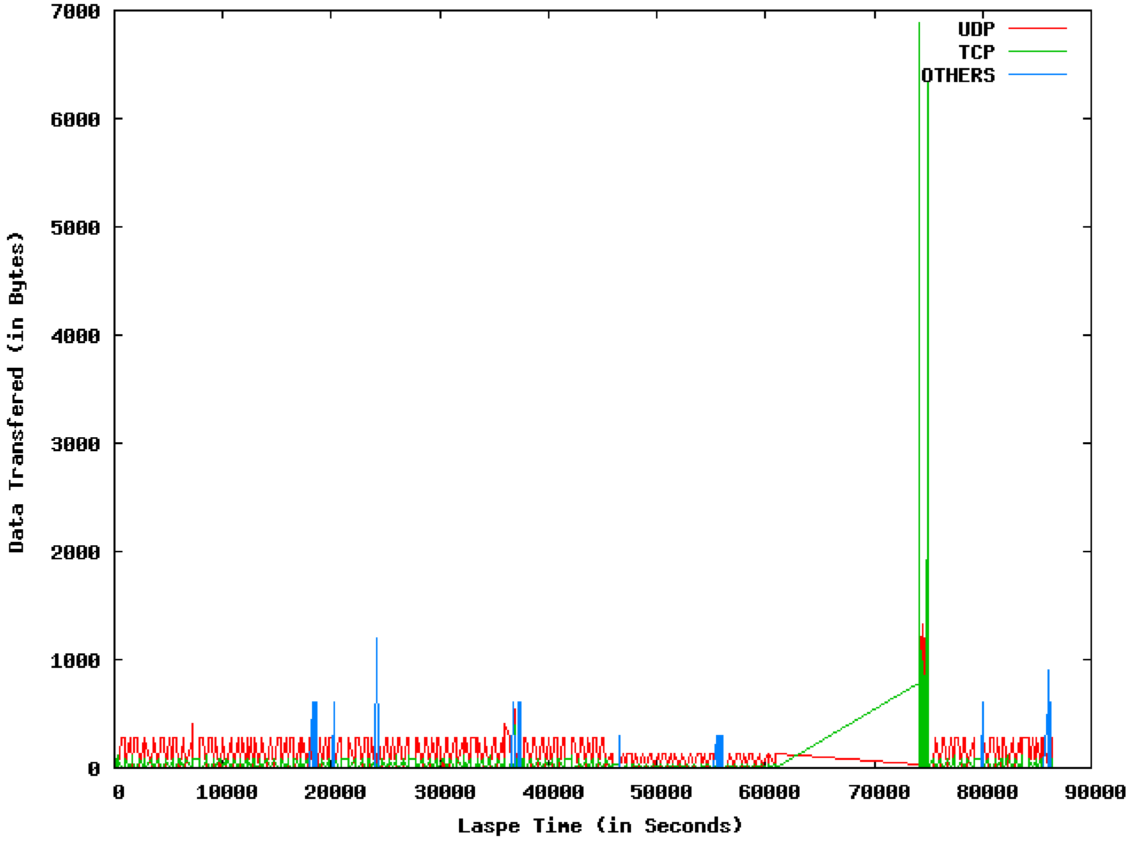


Figure D.45: Mechwarrior: 4 Mercenaries Traffic Chart

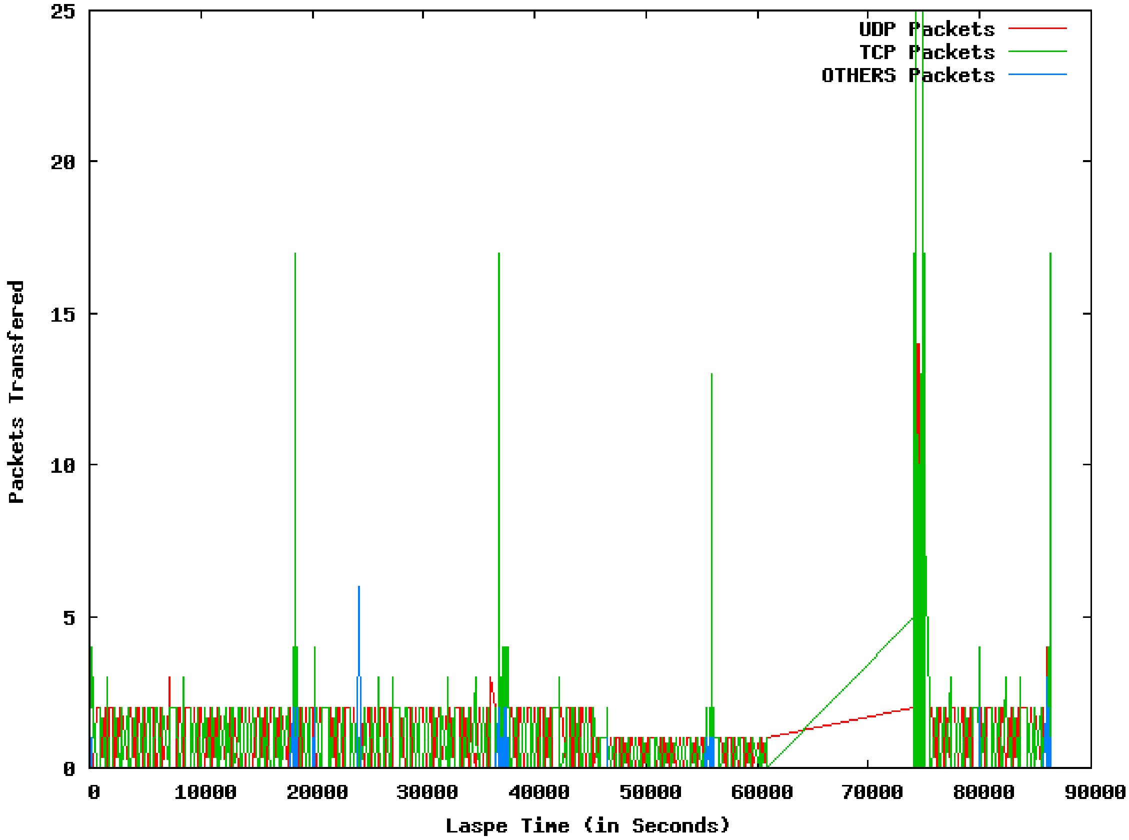


Figure D.46: Mechwarrior: 4 Mercenaries Packets Chart