

AN ANALYSIS OF PLAYABILITY AND FAIRNESS
FOR GROUP GAME SERVER SELECTION

A Major Qualifying Project Report:

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

Yong Heng Wu Fang

and

Nathan T. Tibbetts

Date: April 30, 2007

Approved:

1. Playability
2. Fairness
3. Game Server
4. Group
5. Selection

Professor Mark L. Claypool, Major Advisor

This report represents the work of one or more WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on its web site without editorial or peer review.

Abstract

A study was conducted on an effective way to select a game server for a group of people wanting to play together in a First Person Shooter game using playability and fairness as metrics. The average ping and standard deviation to game servers from the clients are used to calculate playability and fairness for the purpose of identifying suitable servers. Analysis of wide-scale Internet data suggests methods to identify acceptable servers for a group of players.

Table of Contents

1 INTRODUCTION	1
1.1 MOTIVATION	1
1.2 STATE OF SERVER BROWSING	4
1.3 SHORTCOMINGS	6
1.4 LATEST IMPROVEMENTS	7
1.5 GENERAL OBJECTIVES	8
1.6 ROADMAP	8
2 BACKGROUND.....	10
2.1 RELATED RESEARCH	10
2.2 QSTAT	12
2.3 PLANET LAB	14
3 METHODOLOGY	16
3.1 SPECIFIC OBJECTIVES	16
3.2 QSTAT PROBLEMS	17
3.3 GAMES.....	18
3.4 GAME SERVERS	20
3.5 PLANET LAB	22
3.6 PLANET LAB NODES	23
3.7 PLANET LAB PROBLEMS	24
3.8 SOFTWARE.....	25
3.9 DATA FORMAT	28
4 DATA ANALYSIS.....	30
4.1 ASSUMPTIONS.....	30
4.2 DETERMINING GAME SERVERS.....	30
4.3 GAME SERVER ANALYSIS.....	35
4.4 FEASIBILITY.....	63
5 CONCLUSION.....	65
6 FUTURE WORK	68
7 REFERENCES	70
APPENDIX A	71
A1 MAX PING VS. AVERAGE PING GRAPHS	71
A2 MASTER SERVER LIST	72
A3 GAME SERVER LIST	72
A4 CLIENT LIST BY COUNTRY	74
A5 SOFTWARE.....	75
A5.1 Organizer Script.....	75
A5.2 Gatherer Script.....	78
A6 SAMPLE XML DATA FILES	79
A6.1 Master Server Data	79
A6.2 Single Server Data.....	100

1 Introduction

During the past decade or so, with the growth of telecommunications from the introduction of the Internet, the way that people play games has changed. The majority of previous games were single player unlike the now popular multiplayer games using Internet and broadband connections. With multiplayer game options and the Internet, it is possible for many people to play the same game simultaneously. The Internet also allows both people from the same area and different countries to play together.

1.1 Motivation

With the popularity of online gaming most games have a multiplayer component but the number of players varies depending on the genre the game belongs to. There are many different games that can be classified under different genres and each genre has a different tendency towards having a multiplayer component or not. Genres like Action, Role Play Games (RPG), and Simulation Games tend to be single player mainly still. Whereas First Person Shooter (FPS) games tend to be multiplayer, ranging from 2 to 64 players, depending on the map size and the game mode. One of the newer genres, which also has the single biggest multiplayer capacity is Massively Multiplayer Online Games. MMOGs can have as many as ten thousand players in the same server or “world”. From the list of game genres, however, the server selection process is the most important for the FPS games. This is because FPS games, unlike the other genres, focus a lot on player

reaction to situations, which is affected heavily by the performance of the server for the connecting client. The FPS genre, which we will be examining, is one that is characterized normally by having the player in a first person perspective. In other words, in FPS games the player observes the gaming world through the eyes of the avatar. The main objective in FPS type games is for the player to eliminate enemies using an arsenal of weapons. A unique problem, however arises specifically for those who play FPS games as sometimes a group of friends want to play together, but due to being in different locations it is hard for them to find a server that performs well enough for all of them. Finding such a server can be a monumental task due to three factors:

- Number of servers available.

The number of servers available is one of the key factors in the task for choosing a server because it is a limiting factor since the amount of time spent searching an extensive server list can vary. A high number of servers are not always beneficial however; this is because the amount of time needed to go through the servers by trial and error can be time consuming.

- Server performance for each of the players.

Since the number of servers can be large, one of the ways to narrow down the list of possible servers is to filter out servers that do not match the player's desired server performance. To filter, players need to go through the list and check the server performance values and compare those to the player's preferences to be able to filter effectively.

- Coordination of players.

The last factor on the task for selecting servers that can be time consuming is the coordination of players. This is because currently there is no comprehensive tool for players to use that is incorporated into the games to help them coordinate the group as to which server to join. Since the game occupies the entire screen, players are limited in how they can contact each other. Players at the moment have to either exit or minimize the game and use a messenger tool, or use a phone to communicate with each other.

Currently for each of the FPS games out in the market there are a considerable number of servers from which the players can choose. Just going through server lists can be an arduous task because the number of servers can normally range from 50s to 4000 or more servers available per query. Besides having a vast number of servers to choose from, players must keep in mind the performance of the server in relation to them because servers vary a lot depending on:

- Network Performance

One of the factors affecting performance of a server for a specific player depends on the network distance from the player to the server. The network distance of a player is determined by how many networks the player's packets have to traverse, often referred to the number of hops the player has to make to reach the server. The farther away the player is from the server, hop-wise, the more networks they will need to go through. The problem with having to traverse more networks is that there is a chance that there might be a bottleneck in one of the networks. This means that it is possible for one of the networks to be congested with traffic which will make it take longer for the information to pass through such network. It is also possible that the propagation delay might be long due to the distance that the data has to travel. All of this contributes to the latency of

the connection between the player and the server, which is often measured using ping. Ping is measured by the amount of time in milliseconds that it takes a piece of data to travel from the player to the server and back. Like previously stated network performance is important as it might cause information to travel slower from the player to the server, which in turn means bad multiplayer performance.

Lastly the performance of the server can be affected by the players that are connected to and are playing on said server. This is because if there are more players connected to the server then it will take more work for the server to achieve acceptable level of performance. Sometimes a server may try to host more players than it can support and it will cause degraded performance. Also, if there are a few players with high ping then the servers sometimes have to compensate for those players so the server performance will go down since they might have to wait for the player information to reach them before to continuing with the game.

Lower performance can cause players to be at a disadvantage which could make the game less entertaining for all the players on the server. As a group, all the individuals have to consider group performance other before choosing a server, which brings up the last factor, which is coordinating the server search for the group.

1.2 State of Server Browsing

The current state of server browsers is designed to support the individual player. Most server browsers have a simple interface coupled with filters. Current server browsers for games have different protocols for querying and each one looks different

aesthetically. However they are similar in that they provide the ping, the map, the server name, the number of players to the users, and filtering options that handle the preferences of the user based on the values that the server browser reports. An example of a server browser in-game is shown in Figure 1-1. With this information and some trial and error, individuals can normally find servers that suit their needs.



Figure 1-1

The standard steps to using a server browser for the typical player include: select filter options, get list of servers, select a server, connect to server, stay and play game or repeat steps. The first action a player performs in order to play online typically, is to click on the multiplayer option and then choose filters. For example, a player might want to only play on a certain map like de_dust and only look at servers with a ping lower than 300ms. After they set up their filter options they then get the list of possible servers by using the "Find servers" option. Since the filter options have already been chosen, the server browser will automatically narrow down the list to show only servers that match the player preferences. The player then makes a choice from the servers listed on their

server browser screen and connects to the selected server. After connecting to the server, sometimes the player will be in spectator mode and be able to watch the other players play before joining the game themselves. Other times, the player will be dropped into the current round in the game. Soon after joining or after playing a few rounds the player can decide to stay or close the connection to the server and go through the process again to choose another server.

1.3 Shortcomings

The main problem with current server browsers is that they are designed for individual players and do not support server selection for groups. As mentioned above, one reason why current server browser capabilities are not sufficient for a group of players is that there is no easy way for players in the group to communicate with each other to make an informed decision. Currently groups of players would need to either tab in and out of the game to message their friends online through a chat system (like AIM, ICQ, GChat, IRC, MSN) or to call on a traditional phone. Due to the necessity to use outside means to communicate it is hard to coordinate among the players as to which server to join.

Another reason why current browsers are ineffective is that selection must proceed by trial and error. The reason why it is hard to choose a good server for a group of individuals is that each player in the group will have different statistics on the different servers. So while a server might be good for a few people, it might not be as good for all the players, so then as a group it is hard to decide how to compromise. For example

assume there is a group of 8 people and as many as 100 servers possible. For a few of the servers, the ping for 5 out of the 8 people is less than 200ms while for the rest it is higher than 400ms. There are also a few servers where the ping for 3 out of the 8 people is less than 80 but it is around 600ms for the others. Finally there are a few servers that have a ping of 300ms on average for everyone in the group. The question is how to determine which of those servers is best for the group and how fair is it to the group overall. By having to use extra steps to coordinate and test each server individually the time it takes for the group to find a good server can be time consuming.

To coordinate this effort the group of players would have to figure out the ping averages externally. Since there are multiple people in the group, getting this information from each of the players could take up to a minute or more, and testing each server can take several minutes before deciding whether or not if it is good enough. This can take over 5 minutes per server, meaning testing a modest 10 servers can take almost an hour.

1.4 Latest Improvements

Due to the shortcomings in the server browsing process for groups, research interest in this area has been increasing. Some of the newest research on this topic has discovered that manual methods take at least 10 times as long for a group to coordinate than if you use a simple lowest average ping time algorithm[1]. The drawback to the lowest average ping time algorithm is that it might result in unfairness to some of the players in the group. Even a low average ping can have a high maximum ping, affecting

performance for everyone on the server to compensate for the player with the higher ping.

1.5 General Objectives

Our interest in this project lies within the typical situation of a group of friends, who are on separate computers and might be separated geographically and want to play simultaneously, trying to find a suitable server and a procedure on how to find such a server within certain time constraints. We want to diversify the data gathered by extending the number of clients used to gather data as well as having multiple locations to gather data from, to emulate different players. Diversifying the data allows us to characterize the server browsing problem by collecting a large Internet-wide trace to get a representative view of online game players. The simulation of different players will provide us with multiple client data without requiring actual players on the terminals for each client. By further exploring different methods of choosing a server for different players in a group, as well as reevaluating the previous lowest average ping method we hope to find a more suitable or refined method for searching for a suitable server.

1.6 Roadmap

The purpose of this paper is to inform the reader of our research as well as explain in detail to the reader the tools and procedures we used to attain such findings. In the Background chapter of the paper, a detailed report on our findings about current research

regarding FPS games, server browsing, game traffic, player performance analysis, geographic location of servers and players, and how this research applies to our current topic is discussed. Also in the Background chapter some of the tools used for our research will be explained in more detail.

In the Methodology chapter of the paper, specific objectives are explained and how these objectives are accomplished by using the tools mentioned in the background section as well as other tools that were developed for this project. Besides explaining how these tools come together for the use of our research, a detailed report of the limitations and problems that these tools have is offered. After the complete report on the tools, following with a detailed explanation of how the games and servers chosen for the data gathering as well as information regarding on how the gathered data was handled.

After explaining how the gathered data is handled, the Data Analysis chapter explains in detail our findings. Within the details on our findings, the assumptions that we made during our analysis and experiments, possible methods for choosing servers, the effectiveness of such methods, and how feasible these methods are when applied to real world implementation is examined and explained.

With the data gathered and analyzed we offer a summery in our Conclusion chapter. Our conclusion revisits how successful we were in achieving our set goals or why we failed to achieve them.

Following the conclusion, the Future work chapter examines possible topics or extensions to our current research. Since games tend to evolve as systems become more complex research on these methods can be extended to those games.

2 Background

2.1 Related Research

The paper by P. Branch et al[2] provides us with background on the research already done on FPS games. An example of this is that the traffic for games like Half-Life, Counter-Strike, Quake, and Unreal Tournament is similar. The time intervals between the packets of information from the client and the server, as well as the length of the packet are similar for these games. The authors were able to come up with a mathematical equation that can be used to predict some parts of the game traffic like the packet size and the intervals between packets of a game with different amount of players. The results indicate that if a client sent simultaneous queries for these games from the same location to different game servers near each other with similar amounts of players the same ping should be received. Throughout the analysis in the paper it is shown that FPS game traffic has a predictable structure, so then during our data gathering we expect the ping from nearby clients to be similar across games.

The paper by T. Lang et al[3] is a study of the Xbox system traffic while playing Halo as well as a study on creating a simulation model based on the observed traffic characteristics of the system. Analysis of the Xbox traffic in the paper showed that the traffic pattern is simple and depends only on a few parameters. The parameters that affect a packets length depends on are the number of players in the game, the number of clients, and the number of players connected to a client. Both the client and server

packets are sent in an interval of 40ms and depend on the number of players connected to the client or server. The result of the study was a NS2 model for the Xbox system that could be used to evaluate network suitability for multiplayer games. The results from the study are important to our work, as [3] extends some of the findings presented in [2] and shows that FPS games have predictable traffic structures that are similar. This means that we can expect the traffic data gathered to be similar in many respects as they will most likely share similar traffic structures.

M. Dick et al[4] discussed a playability metric through the use of ping along with standard deviation to use as a fairness metric. This makes it possible for us to use playability and fairness as metrics also for determining how suitable a server is for a group of players. The ultimate results in the paper were that different games have different tolerances for latency and jitter before they affect the player's perception. For example, for the FPS game Unreal Tournament latency seems to be the most affected when it is higher than 150ms (milliseconds) and is unplayable at over 500ms, while Counter-Strike players do not seem to be affected too adversely with pings under 500ms. This is remarkable because it indicates that playability and fairness definitely affect a player's ability and that they are able to determine some values in context of playability for FPS games. For example, UT2004 is adversely affected at over 150ms but Counter Strike is not adversely affected with pings up to 500ms.

C. Chambers et al[5] focus on tracing and gathering information about number of servers and the localization of the player base for some of those servers. It infers that a majority of servers are located on the northern parts of the hemisphere and that the player server preferences are mainly by the time of the day and the location of the player.

Player activity peaks from 8pm until 4am local time. For our work, the time frame in which there seems to be the highest amount of players, providing the best time to capture a snapshot of traffic for FPS games at peak times.

S. Gargolinski et al[1] show how a FPS game would look from two different perspectives with friends that are trying to play together in a suitable environment. The problem tackled in the paper is how the players would go about looking for a server that was good enough for both players since data gathered from the server browser would look rather different for each of them. In essence our project is an extension over this previous paper, since there are overlapping features. We are taking snapshots of how the games would look like for multiple people (not just two or four) around the world, and gather more data from a bigger range of games.

2.2 Qstat

We researched a tool called Qstat that provided us with a way of querying servers from various different game types without needing any of the respective games[6]. Qstat is a program that can be used to gather data on servers that are listed on a master server service typically offered by the developers of an FPS game. A master server is a server normally offered by game developer to keep a list of the individual game servers available for players to join and play. The list of servers contains the IP address of the servers as well as their status. The status for the servers can be up, down, or timeout. If the server is up then it is online and responding to queries, if it is down then it means that it has officially stopped its services, and if it is timeout then the server is not

acknowledged as down but there is no current response from that server. The software is effectively an all-in-one game server browser, allowing us to query most of the online games that use a traditional server browser for choosing a server to play on. This includes older games such as Quake 3, Tribes, Counter-Strike, Heretic II and newer games such as Quake 4, Doom 3, and Counter-Strike: Source. We researched Qstat's various capabilities and limitations to determine if this would provide us with the data we would need to meet our goals.

After going through the accompanying documentation and trying some manual queries to test its functionality, we found that it provided the necessary information we wanted. With the right options we could query a game server and have the software return the information a player would normally receive when viewing a server's information in the in-game server browser. The data gathered on servers from Qstat provide information on:

- The name of the server
- The IP address of the server
- The ping of the server
- The number of current players connect to the server
- The maximum number of players that can be connected to the server
- The current map being played in the server
- Specific rules of the server
- The ping of the players currently connected to the server
- The score of the players currently connected to the server

Qstat is a command line application that can query servers of different games. In case that an extension or changes need to be made, Qstat source code (written in C) is readily available for making these changes. Also since Qstat is a command line application rather than a GUI oriented application, it is suitable for automating for the use of data gathering in large quantities. The results produced by Qstat can be displayed on screen or written to a file in different formats like HTML or XML. For our purposes, we used XML to make it easier to parse through later for our data analysis. This was useful for our analysis because of the fact that we could have the data received as a result of a query stored in an XML file, making the data easier to parse into a more suitable format or filtered down to key elements from the information gathered that we were interested in looking at. This required writing a script or program to perform the translation, but was trivial given the many tool sets available for reading and parsing XML files available for most programming languages.

2.3 Planet Lab

Another tool or resource we used for our research is Planet Lab, which is an international network of computers [7]. Planet Lab hosts a minimum of two computers per site, with each site being a part of university, organization, or corporation. The Planet Lab network's purpose is to provide multiple computer resources for researchers to use for networking across the globe. This collection of machines distributed over the world serves as a research and testing tool for any group that has contributed a system to Planet

Lab. Contributing a system requires setting up at least two computers that have access to and are accessible from the Internet. Each system that is part of Planet Lab is referred to as a node and runs a Linux based Operating System that provides secure and completely separated virtual machines for each Planet Lab user that uses the node. After the nodes are part of Planet Lab, at least one person is identified as a Principal Investigator (PI), whose job is to over see and manage the people who are to have access to Planet Lab as part of the group that contributed nodes. At this point people can sign up for and get approval for accounts that will let them manage what nodes they want to have access to for whatever research purpose they have. A user then has to generate and upload the public part of an SSH key pair that is used to authenticate and encrypt the connection made to Planet Lab nodes. Once the public SSH key has propagated to all of the users nodes, the user now has access to the nodes they chose through an SSH connection. Planet Lab is covered in more detail in 3.53.5 below.

3 Methodology

3.1 Specific Objectives

Our objectives for this study included developing a system of simulating numerous players querying game servers at the same time. The objectives also included using this system to gather data on the queries made from these simulated players and analyzing this data to attempt to determine the best strategy for picking a server for all these players to play on. Ultimately, this could be used to design a better game server browser that would facilitate coordination between multiple players wanting to play on the same server.

Specifically, we used a method for querying game servers in a manner similar to how an actual player, in the game, would query these servers. We also determined a way of making these queries from different physical locations so that we had realistic data on what different players in different places would see when trying to play with each other. With such a system in place, we then devised a way to organize and analyze the data gathered. Finally, we needed to determine some possible methods for picking a best server for a given set of players to play on and investigate how well the methods work.

To accomplish these objectives we looked into a game server browsing tool called Qstat (first mentioned in Section 2.2) that allowed us to simulate a server browser query, and a network of computer systems called Planet Lab (mentioned in Section 2.3) that would allow us to make these queries simultaneously from different places across the world. To accomplish the other aspects of these objectives we used the scripting

language Perl to combine these tools into the system of emulated players located across the world to gather the data we needed. We also used Perl to help parse the resulting data files into the file formats that we needed to analyze and determine the best method for picking a common game server for a group of players.

3.2 Qstat Problems

Despite how useful Qstat is, it was not without its problems. One problem we came across was not a direct reflection on Qstat as a tool. The issue was that not every Linux distribution that provides Qstat as a package that can be installed via that distribution's package manager packages it with the same files. More specifically, when testing on local systems and compiling it from the source directly from the tool's web repository, Qstat appears not to support querying any game based on of the Half-Life 2 game engine, namely Counter Strike: Source. Whereas when installed on one of our personal systems which is running the Gentoo Linux distribution, the package provided has an additionally configuration file that allows Qstat to query games based on the Half-Life 2 game engine.

This was mainly an issue when we determined the games we would query because we had produced our game list based off the games that could be queried from the personal system. Then when we went to develop the script built around Qstat, we were testing with the school systems and getting errors about not supporting Half-Life 2 based games. Once we had determined it had something to do with the version packaged with Gentoo, we still spent time trying to determine what was different about the version packaged with Gentoo compared to the one taken directly from the Qstat web repository.

Eventually we found that it related to a configuration file that is included from some other source that provides this additional support. This gave us the option of considering Half-Life 2 based games when choosing the games to query.

The biggest issue, however, with this tool is the fact that it is not updated often. This is a problem because if any of the games it currently supports changes its server browsing protocol, Qstat will not work with this game until it is updated to handle these changes. What this meant for us was there was a chance that for any of the games that we choose to query with this tool, the protocol that was being used might not be the most recent one or the most effective. This could result in slow queries, not getting all the information we needed, or not being able to query some game servers. This was highlighted by the lack of proper support for Half-Life 2 based games so that it currently requires a separate file to provide the needed support.

3.3 Games

Finding that Qstat provided us with the interface to querying game servers, we were left with deciding what games on which to focus. Qstat supports querying approximately 50 types of games, but collecting data for all these games would create a problem. Assuming we were going to query every type of game that Qstat supports and we were to use 20 clients querying every 15 minutes, this would result in approximately 15000 data points for only one day of querying. Each of these data points would include the ping from the client to the server, the number of players on the server, the maximum number of players allowed on the server, the names of all the players on the server, the scores and deaths of each of the players, and each players ping to the server. Collecting

data on all of the games Qstat supports clearly would result in a large amount of data that would, in the scope of our project, make it infeasible to analyze it all.

This made it necessary to concentrate on a subset of the games Qstat supported so that the resulting data was manageable. Looking at the list of games that Qstat supported, we decided to gather data on only First Person Shooter (FPS) games, since that was the main game type that Qstat supports. Our next criterion for trimming down the list of games we would query was to pick the newer games since these would be the games most players are playing. Using these criteria left us with Counter-Strike: Source (2005), Quake4 (2005), Doom3 (2004), and Unreal Tournament 2004 (2004). However, we researched what master server each of these games uses to get the latest list of servers and could not find a master server for Unreal Tournament 2004 that listed any server. This meant that we wouldn't be using this game since we had have no way of compiling a list of servers to query.

Despite the fact that newer games we are generally more popular, not everyone plays the latest games available. People just prefer to play older games or older versions of their favorite game instead of the newest. Also not everyone has a computer that can support playing the latest in bleeding edge video games. So we decided to also include older iterations of the same game when possible or at least an older game by the same company. This brought our list of games up to 6 games with Counter-Strike: Source, Counter-Strike (2005), Quake 4 (2005), Quake 3 (1999), Doom3 (2004), and Return to Castle Wolfenstein (2001). We felt that this gave us a good variety of games and gave us a few different points of comparison if we decided to compare any aspect of the data of the games.

3.4 Game Servers

For each game, we needed to determine specific servers from which we were going to gather data. This was necessary because we had found when we were experimenting with Qstat that any servers that were not up and were not coming back as down would eventually time out. The problem was this took a long enough time that given enough timed out servers one set of queries for a game would take too long. For example, if it takes Qstat 10 seconds to return a timeout, and for a given master server there are 8000 servers listed, and 30% of them were going to timeout, this would result in the query of every server taking 24000 seconds, or 6 hours and 40 minutes. For the sake of getting enough query sets in a day we would need to make sure the queries would go as quickly as possible and this would be best achieved by querying a set list of servers that were most likely to be up and remain up.

In order to accomplish this we needed to collect data on the various individual servers in a similar fashion to how we would query them individually. We would then look at which servers were as close to up 100% of the time as possible. So we set up a script to run from a local school system to query only the master server for each game, as shown in Appendix A2, wait for 30 minutes after querying, then query again. We assumed that there would be some variation in the number of servers available during the week as compared to the weekend. Also, that there would be a higher demand on game servers during the weekend thus potentially causing servers to fail. For these reasons we ran the script for each game for approximately a month in order to gather a month's worth of data to compare for any of the trends we assumed would occur.

Once we collected the data, we had to parse out the relevant aspects of the data into their files so that we could then determine how often the servers were up. We wrote a script to read through all the XML files and pull out the following information:

- Date and Time of the Query
- Server IP Address
- Server Name
- Number of Players Currently on Server
- Maximum Number of Players Allowed on Server
- Server Status

This data was put into a spreadsheet file for each time the server showed up. This created its own problems though, there were so many unique servers and we had taken so many queries that the information would not fit in one spreadsheet file and still be able to open the file in a spreadsheet program since the software limits the file to approximately 65000 rows. So we had to organize the file by IP address of the servers and then split it up into smaller files that could be analyzed. For the purpose of determining which servers to query, we then took the result of dividing the original spreadsheet file up and pulled out the total number of times a server showed up based on unique IP and how many of those times the server was up. Also, since we did not want to query just the servers that were up the most but also make sure we ended up with servers that had empty space for additional players, but at the same time were not completely empty. This was done to get a more representative range of servers akin as to what a player would observe from a query.

Considering the total time to complete one set of queries and strike a balance between not having too much data and having too little data, we decided to choose 20 servers for each game. We had considered using a percentage of the total number of unique servers per game to account for the fact that some games, due partially to popularity, partially to how old a game might be, would not have as many servers as other. However, we opted out of this since it would make doing any comparisons between the games difficult since we have a different number of servers for each game. Also, we had considered this option because we were unsure as to whether or not certain games would have enough servers that were up consistently enough to have enough servers to query. However, this concern was alleviated after looking at the sheer number of servers all the games had, and the fact that a large number of them were up almost 100% of the time. We then chose the 20 servers for each game based on the criteria, wrote them down, and moved on to the next task. The 20 servers for each game are listed in Appendix A3.

3.5 Planet Lab

After identifying a tool for performing game server queries and identifying the games and respective game servers that we were going to query, we were left with finding computer systems to run these queries from. We had initially considered obtaining online access to any computers that were attached to the Internet from friends and family that we could use for our data gathering. However, constant access to these systems provided little to no guarantee of remaining active, and it also limited by how spread out the systems were by who we knew. Wanting to avoid this scenario if possible, we researched a world wide system of computers called Planet Lab.

Planet Lab was just the tool we needed for simulating queries of users located around the world. Even better was the fact that the process of getting computers and setting them up as nodes for Planet Lab had already been started. Once we had our accounts, we began picking nodes and setting them up so that they had all the software we would need installed. We needed to install GCC for compiling Qstat and Perl modules, FTP for retrieving Perl module source code, unzip for unpacking Perl module source code, Qstat, the Expect Perl module for our scripts, and our own custom made script.

3.6 Planet Lab Nodes

At the start of our experience with Planet Lab, there were approximately 750 nodes provided by universities and companies from almost every continent. Our method for choosing nodes was very simple; we wanted the majority of them to be located in North America, more specifically in the U.S., then we wanted to have at least one node from the other continents, but hopefully at least one from the larger countries in each continent. Since Planet Lab as a system is under constant development and since intermittent problems can affect any given node, not all of the nodes are usable all the time. Luckily a tool, called CoMon [8], was created that provides a mean of checking on the state of any given node without having to add it to your list of nodes, wait for your SSH key to propagate, and then test to see if you can log into the node. Using this tool, we identified universities that were in the areas of the world that we were interested in and had at least two nodes that were listed as functioning, and added them to our list of

nodes. By the time we had identified all the areas we wanted nodes in and added any nodes that we could find, we had a list of 45 nodes to run queries from.

At this point we could start setting up the nodes so that they had all of the software that we needed for gathering our data. This included downloading and compiling Qstat, installing various dependencies for our script as described in the Software section below, and making sure any folder structure was set up the way our script was expecting it. It was during this process and the testing that we conducted over the next few weeks that pointed out some of the nodes that we had chosen had become unusable whether because they were completely down or we were told we did not have authorization for the node. This resulted in our list of selected nodes being reduced to 25 nodes, which are listed in Appendix A4. Once the software had been tested and everything worked, we set the whole process in motion and closely monitored out nodes to watch out for any failures or errors we missed in testing.

3.7 Planet Lab Problems

Planet Lab ended up being the tool we needed to get the properly spread out and diverse clients that we were looking for, but the system itself presented problems. The first major problem was the dependability of the nodes, or at least the ones that we chose. While we did not expect all of the nodes that we chose to be up and available for the entirety of our software testing and data gathering process, we also did not expect that we would end up not having use of almost half of the original set of nodes that we had chosen. This was only exacerbated by another problem, a lack of node status. While the tool for seeing the status of the various nodes, it was not perfect and tended to be outdated by at least a few hours. This lack of dependability was worsened even further

by an unfortunately timed upgrade to the entire Planet Lab system which caused nodes to be unpredictable taken down for upgrade, and then brought back up. In addition to the inconvenience of having a node down, when the node came back up after an upgrade, everything that was installed and any data that had not been retrieved was deleted.

While dependability in general was a problem, it was not the only issue we had. The virtual machines that we as users had access to had a distribution of Linux installed called Fedora Core. Fedora Core 2 was the version that was installed initially before any nodes underwent an upgrade, which is the version that was released over two years ago, and has already been surpassed by four newer versions. The fact that this Operating System was out of date is highlighted by the fact that Planet Lab had to have their own repository for Fedora Core 2 packages because there are not any repositories that still bother to carry Fedora Core 2 packages. This in and of itself caused another problem we had - incredibly slow installs through Fedora's package manager. This was a direct result of the fact that regardless of the location of the node, it was configured to get its packages from the Planet Lab repository located on the west coast of the United States. This resulted in very long download times that suffered numerous dropped and timed out packets, and in the worst case even failure to even download the package. Once everything was installed, though, this was only a problem if a node went down and came back up without the software we had installed.

3.8 Software

The key component that made this entire data gathering process work was the software that we wrote to actually coordinate and perform the queries. We divided this

into two major components, the part that gathers the data on the clients and the part that coordinates all the clients so that they perform queries at roughly the same time.

Since Qstat is depended on by our software, we had to provide a Qstat executable on all of the nodes so that the Gatherer script could execute queries on the game servers. This was done by simply ensuring all of the tools needed to compile a program were installed, which included gcc, make, and various gcc libraries. Then the source was downloaded from the Qstat web repository, unpacked, and compiled.

The part of our software that actually performs the queries and collects the data we called simply the “Gatherer”, shown in Appendix A5.2. This part consisted of a Perl script that would take in a game type, list of game servers, a date, and a time. It then used all of this to construct the appropriate call to Qstat for the given game type and make sure the name of the output file was formatted properly with the game type, game server, client name, date and time in it to make each output file unique and readily identifiable.

The second part that coordinates all of the queries was named the “Organizer”; shown in Appendix A5.1. This part also consisted of a Perl script that would take in a filename for the file that contained the list of nodes it was to connect to, a game type, and a list of game servers. It used all of this to form connections with all of the clients and tell them to execute their Gatherer scripts, providing those scripts with all of the information they needed. It would then monitor for them to finish executing, and then tell all of the clients to send back their data using SCP. This was all accomplished using a Perl module called Expect that makes it relatively simple to interface with command line tools and in essence write programs as if you were typing the commands in yourself and checking for certain output.

All together the gathering process proceeded as follows for each game; The Organizer would be started on a local central system. The script would initiate an SSH connection with each of the Planet Lab clients. Once all the SSH connections were made the Organizer would then tell the clients to execute the Gatherer script, providing the game being queried and the list of game servers to query. The Gatherer script would then iterate over the list of game servers and query each of them using Qstat, storing the resulting response into an XML file. While the Gatherer was querying game servers, the Organizer monitors the clients for the Gatherer script to finish. Once the Gatherer script is finished, the Organizer uses SCP to transfer the files to the local system. For each game the organizer script was set up in a cron job to query every 30 minutes, with each game offset from the one before it by 5 minutes to evenly distribute the query load. This entire process is represented in the diagram in Figure 3-1.

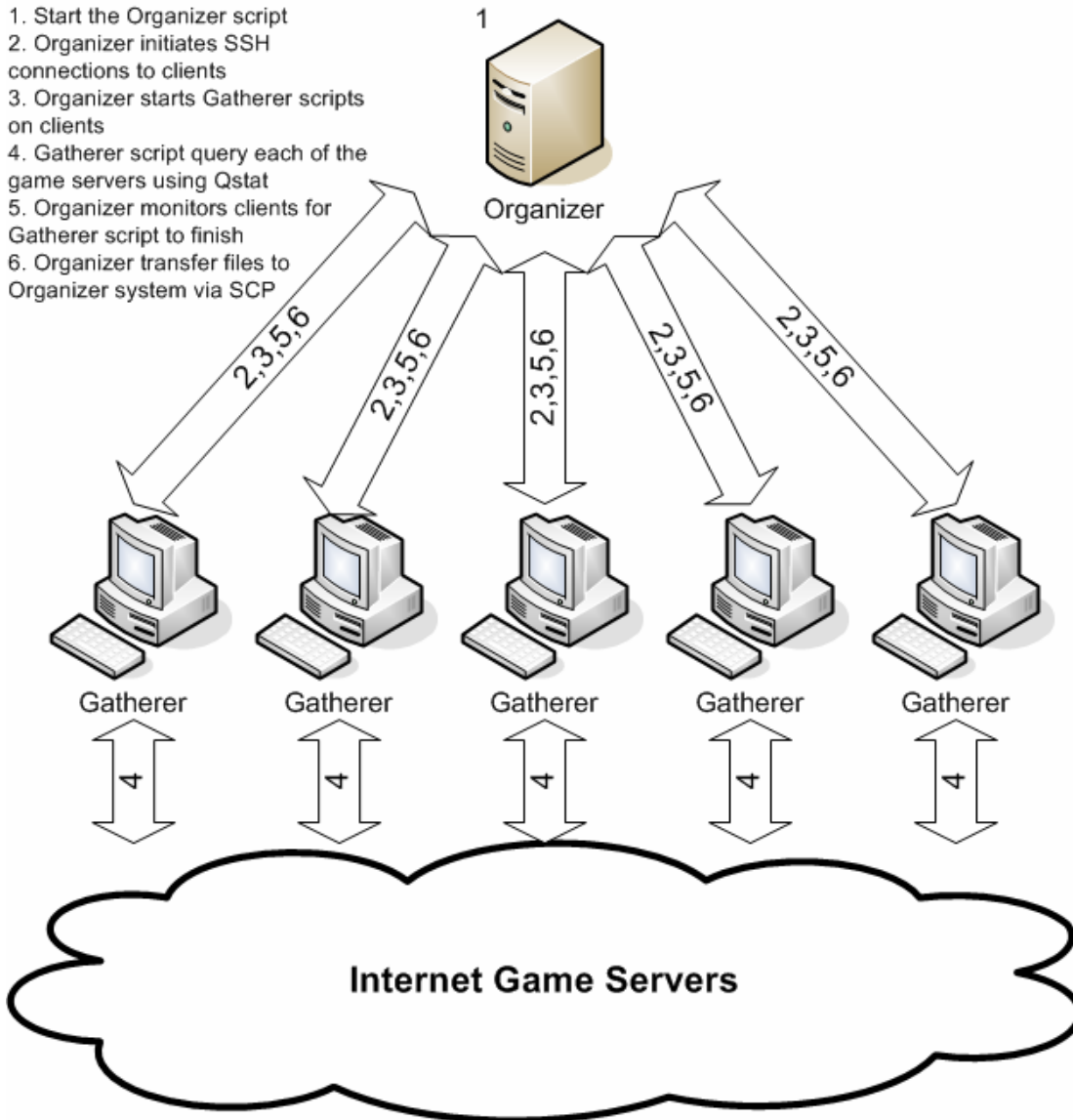


Figure 3-1

3.9 Data Format

The format for the gathered data takes the form of an XML file, which made it easier to parse for relevant information. There is a difference in the XML file for the master server gathered data and the single server gathered data. The master server data XML files were much larger and contained basic information on all the servers that were retrieved from the master server list. The basic information entails server name, server

address, server ping, current number of players, maximum number of players, the type of game, and the current map being played on. The single server data XML files were much smaller as they are specific detailed queries to the servers we had chosen. The information in these XML files contained the name of the server, the address of the server, the game type, server ping, current number of players, maximum number of players, the name of the players, the score of the players, the ping of the players, the map being played on, and the rules enforced on the server. Example XML files for both master server and single server queries are shown in Appendix A6.

Due to the scope of our project and interests we did not use all the data we gathered. For example we did not make any use of the server name that we retrieved as it did not offer any insightful information on how to choose a “best” server. The data that we focused on was varied depending on if the data was from the master server set or the single server set. For the master server set of data we were mainly interested in the server address, the number of players on in each server at time of query, the maximum number of players for each server, the date and time of the query, the status of the server, and the ping of the server. For the single server data set we were interested in the address of the server, the date and time of the query, and the ping of the servers from all the different clients.

Since we had chosen to use Excel or a similar spreadsheet program to examine the data we needed to turn the XML file along with the relevant data that we wanted into a comma separated file. Due to this, we made a script to parse through the XML file and generate a comma separated file with the data that we wanted extracted from the XML file.

4 Data Analysis

4.1 Assumptions

Due to lack of knowledge on player preferences when it comes to choosing game servers, assumptions are necessary to narrow down the factors affecting our analysis. Therefore we assumed that players in a group look for the lowest ping, a certain number of players in the server depending on group size, and that all the players can get into the servers. The possible problem with player connections to a server might be caused by some servers having an upper limit on the ping of players allowed to connect to it. For example it is possible for someone in the group to have a ping of 600ms but the server might only allow a max of 300ms, thus that player would not be able to connect. We are assuming that such a case does not happen.

For any of the query information to be valid, the amount of time required to decide which server is “best” needs to be short enough that the network and server conditions do not change significantly. The reason for this is that if the process takes much longer then the data gathered about the servers might not be accurate anymore especially when considering servers with population in the server already.

4.2 Determining Game Servers

The data gathered from the master servers of the three different games is important as preliminary data for our analysis. By using this data, we can get a better

understanding of the games before conducting an in-depth analysis. Looking at the scope of the project, there is a large amount of data that can be gathered as mentioned in Section 3.4 and due to the limited allotted time we decided to use a subset of the data. To determine the subset of data to use, we gathered a month's worth of data from December 1st to December 31st of 2006. The difference between "Down" and "Timed Out" for the servers is that if it says it is "Down" then that means that the server let the master server know beforehand that it was stopping its services but if it did not inform the master server then it would "Time Out" when master server checks status. All this matters because the master server query depends on the data stored in the master server since Qstat does not query all the servers itself when doing a master server query. Examining the data from the master server on the games for Quake3, Quake4, Doom3, and Return to Castle Wolfenstein, we determined that the number of servers per game to monitor should be a set number rather than a percentage of the total servers.

Game	Servers listed (average)	Servers Down (average)	Servers Timed Out (average)	Servers Up (average)	Standard Deviation of Servers Up
Quake3	553	4 (0.72%)	28 (5.06%)	521 (94.21%)	21
Quake4	1642	91 (5.54%)	154 (9.38%)	1397 (85.08%)	156
Doom3	76	1 (1.32%)	8 (10.53%)	67 (88.16%)	8
RW	161	1 (0.62%)	1 (0.62%)	159 (98.76%)	13
Total	2432	97	191	2144	198

Table 4-1

Since the number of servers for each game varies we needed a way to take the variance in servers into account when deciding the number of servers to query. In order to do this we took the average and standard deviation of the number of servers over the

course of the month. Looking at the number of servers listed per query in Table 4-1, it is obvious that there is a large disparity in the number of servers for different games. Due to the disparity in numbers, taking a percentage of servers would make it more difficult to compare accurately. For example, if we decided to use 10% of the servers then for each of the games we would have 160 servers for Quake3, 55 for Quake4, and 7 for Doom3. The numbers in Table 4-1 represent the average number of servers per query obtained during our month long data gathering session. While the number of servers for Quake3 and Quake4 might be acceptable for comparison it would be hard to compare Doom3 and Return to Castle Wolfenstein (RW) since the number of servers from them is very small and might not be an accurate representation of the data. Therefore, we chose to pick a set number of servers, 20 per game, shown in Appendix A3, based on the standard deviation calculations. Examining the smallest number of servers available per query, which was 74 from the Doom3 game, we took two deviations from the mean to account for 95% of the possible range of the total number of servers online. By taking two deviations, the number of servers for Doom3 would range from 54 to 94, but since we are limited by the smallest number of servers in order to guarantee consistency it limited us to a maximum of 54. Since we wanted to further ensure that the servers we picked remained constantly online we chose to work with a set of 20 servers per game.

After narrowing the number of servers we were going to use to gather information, we needed to choose the specific servers we would monitor over the course of several weeks. The servers we monitored needed to be stable so that the data we gather does not have points in the data missing. Besides needing stable servers, we needed to find a representative range of player populations on the servers.

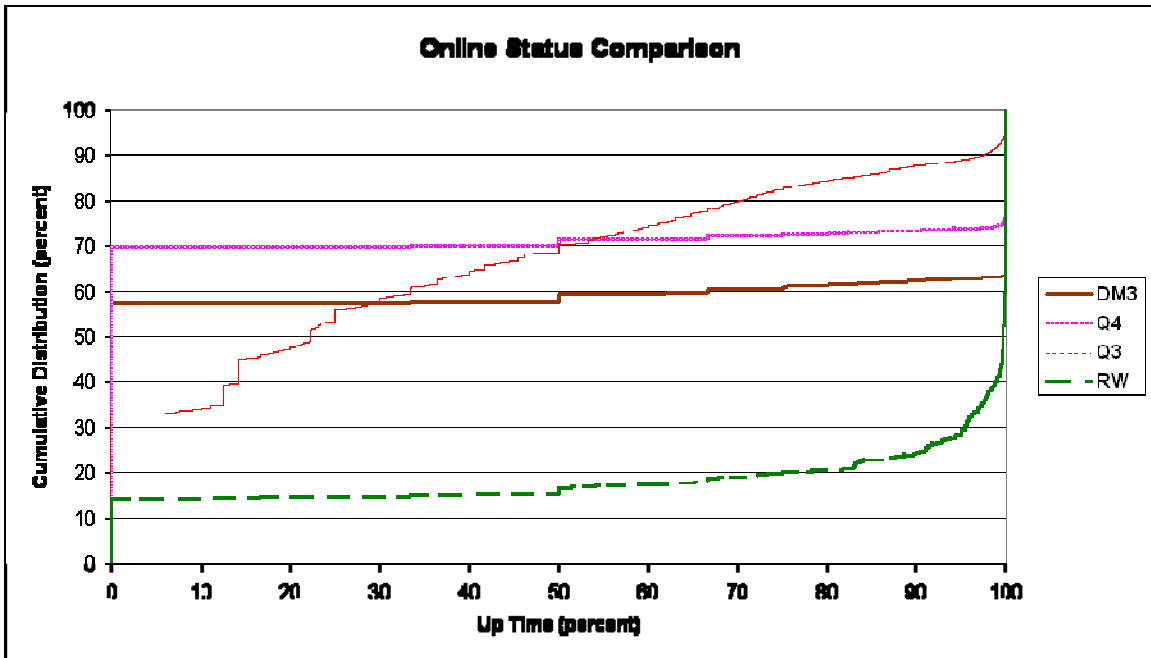


Figure 4-1

The values in Figure 4-1 are computed from the list of all the servers found during our month of data. Examining the data we gathered from the master servers we were able to determine that for most of the games, Quake4, Doom3, and RW, the servers can be generally categorized as timed-out/down all the time or up all the time. Observing the graph we can see that for the games Doom3 and Quake4, 58% to 70% of the unique servers listed during the whole time we queried them were either timed-out or down. The list of servers is the one acquired from querying the master server and included in that list were servers that during our queries were down or timed-out. However, we kept those servers as it was probable that they did not respond to our clients or came up for a very short duration. This seems to be the case for those servers as they appeared in our queries for 1 query up to 500 queries, with 500 being the highest number of times a server showed up before it stopped showing up on the list of servers. Normally even if the server was down or timed out during some or all of the occasions, it would still be listed,

and it is highly doubtful that it was due to the master server not updating the list enough. Also the jump from 0% to 100% is rather flat, which means that very few servers are down or timed-out for a small portion of the time. Return to Castle Wolfenstein is similar but the line curves when the servers are up 80% of the time. This means that for those 3 games that most of the servers we choose were up almost all the time, better guaranteeing that the service will not fluctuate. The only game that does not follow this trend is Quake3, which indicates that a lot of the servers fluctuate between being up and down. Therefore, the servers we choose for Quake3 were up over 80% of the time to keep consistency.

Continuing to narrow down the number of servers that are candidates for our monitoring, we sought a representative range of player populations on the servers.

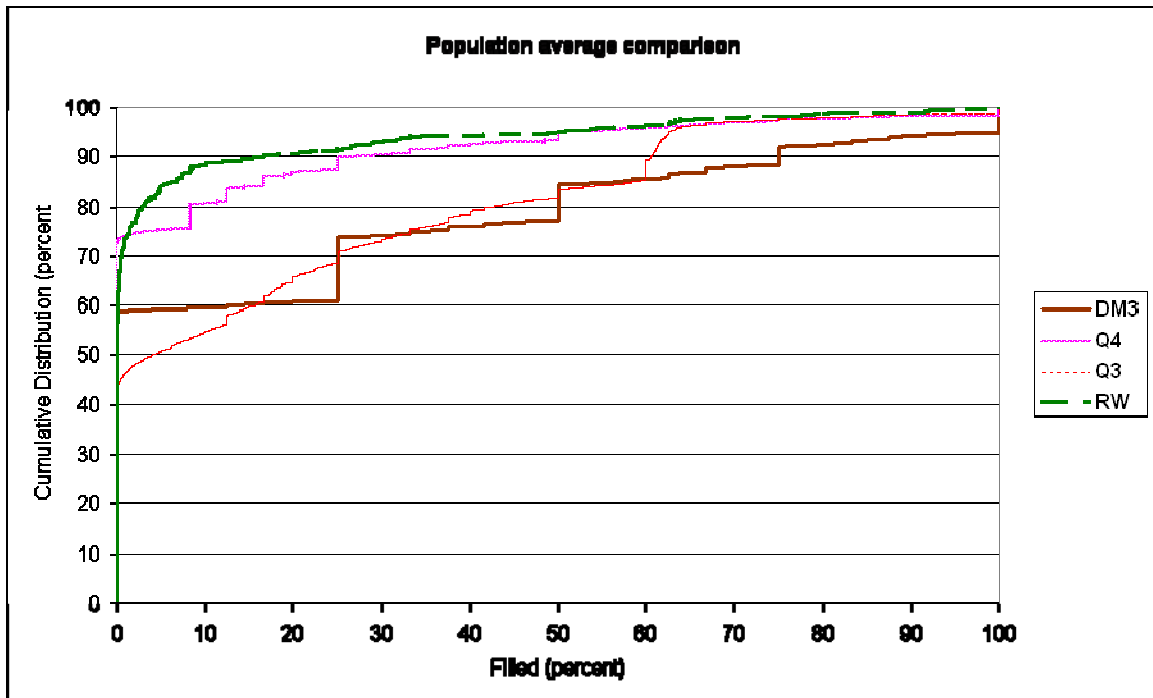


Figure 4-2

To do this we examined our gathered data about the population on the servers at the times we queried. By looking at the graph presented in Figure 4-2 above, we can determine that the majority of the servers are mostly empty. The population represented in Figure 4-2 is compromised by the population from all the servers listed from the master server. Therefore we have to be careful when choosing servers. Since we want our servers to be representative of the server population we chose a few servers that do not have players at all and some that ranged from having a few players to having many players most of the time. The list of the servers used for the single server query datasets are provided in Appendix A3.

4.3 Game Server Analysis

While players often play video games quite often, most do not play every day and definitely do not play the same amount every day due to various real-world activities. This is clear from the fact that even the most devoted gamer has other responsibilities to attend to. This can include going to school, completing daily schoolwork, and working at a job. While this is not an exhaustive list, it still highlights the fact that players do not play all the time. These various other activities also occur at roughly the same time of day for those located in the same part of the world. This would seem to indicate that some days would cause game servers to be busier than other days.

Using the data we gathered from querying the game servers over a month, we looked at the total number of players that were playing on all the game servers listed by the master server. This allowed us to look at any potential patterns in gamer activity over

this period. This data for the month is shown below as raw data for Quake3, Quake4, and Doom3 in Figure 4-4, Figure 4-6, and Figure 4-8, respectively. The same data was run through a moving average of 24 points in order to smooth out the data and to highlight the overall long-term trend in the data and is show in Figure 4-5, Figure 4-7, and Figure 4-9.

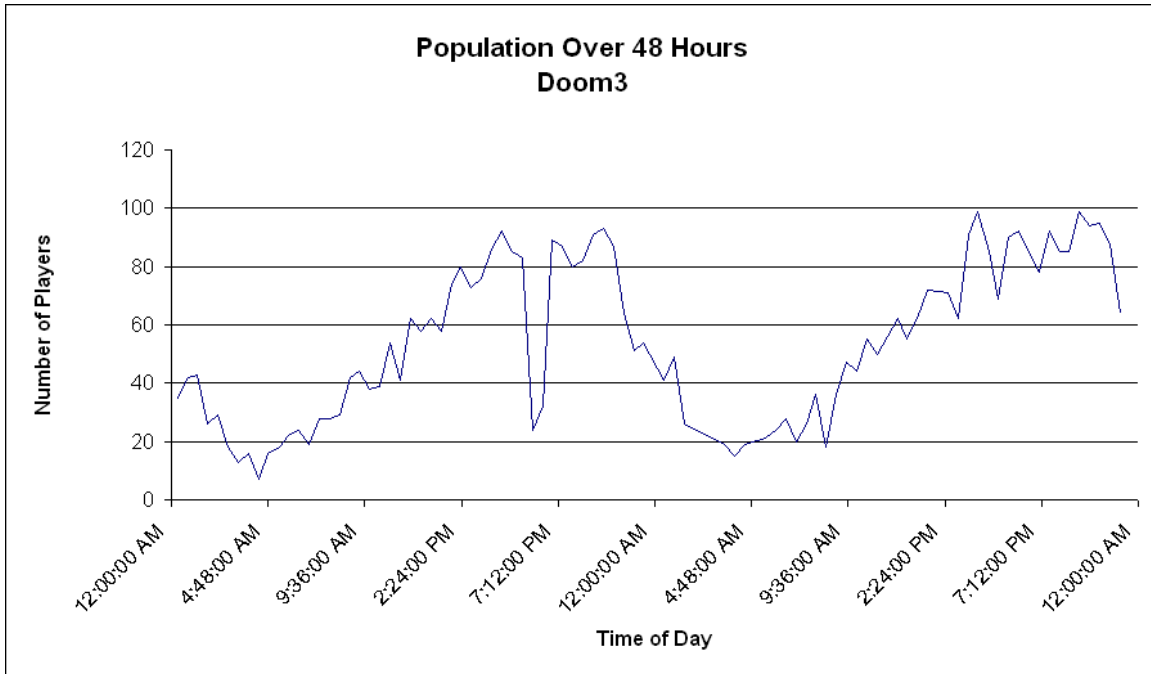


Figure 4-3

Looking at just the raw data in Figure 4-4, Figure 4-6, and Figure 4-8 it is obvious how in one day there is a cyclical pattern as people come home from school and work, and start playing games, bringing the population total up towards its peak. A closer view of a single day can be seen in Figure 4-3. Then as it gets later at night, players start going to bed and the population starts dropping until it reaches its lowest at approximately the middle of the day. Finally, the cycle starts over heading into the afternoon and early evening when gamers become more available again.

What is not so apparent in Figure 4-4, Figure 4-6, and Figure 4-8 is the slight shift the daily cycle makes over the course of a week. The cycle is typically at its lowest on Monday and increases as the week goes on, increasing the most going into the weekend, then drops back down going into Monday. This shift is more visible once the data has been smoothed and the data is viewed on a smaller range, as shown in Figure 4-5, Figure 4-7, and Figure 4-9. However, despite the fact there is an overall increase in the number of players playing over the week, it is not significant enough to consider when trying to pick a server on any given day.

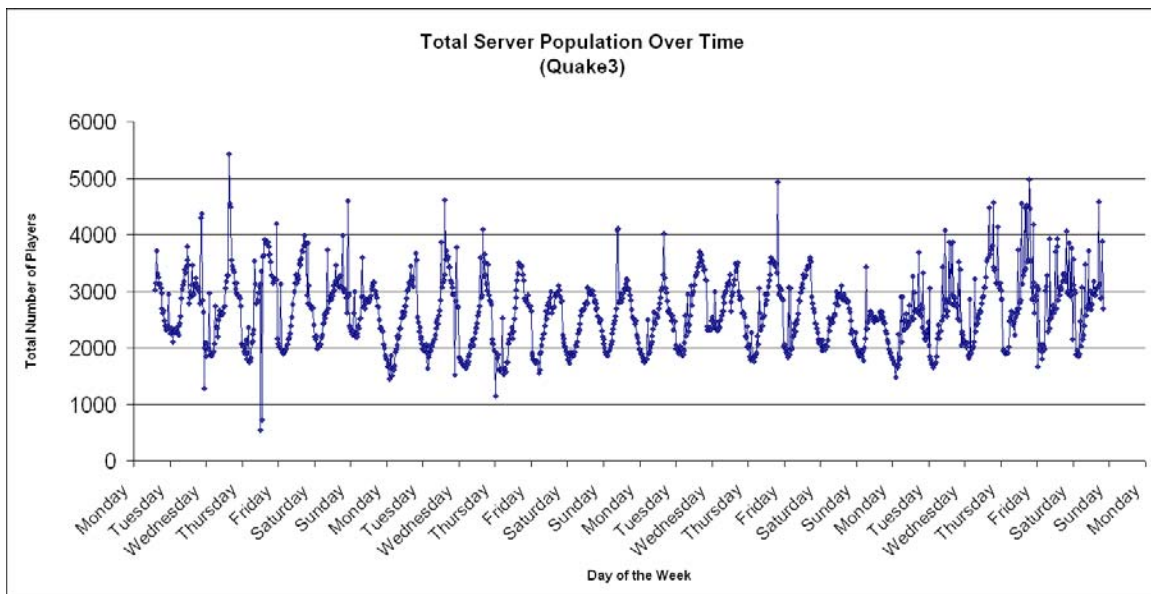


Figure 4-4

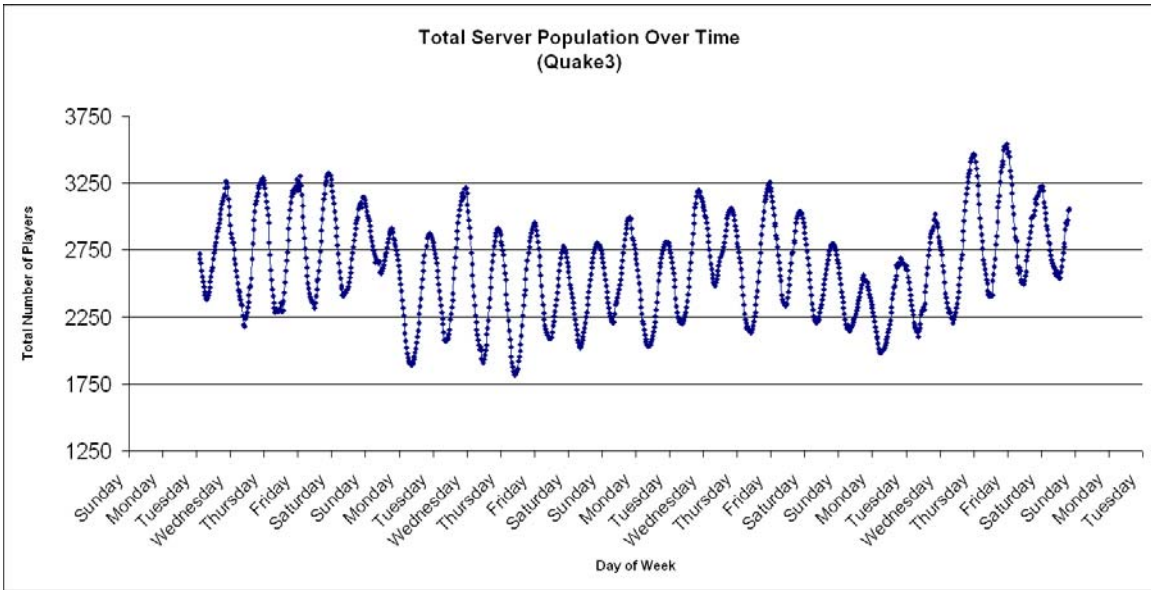


Figure 4-5

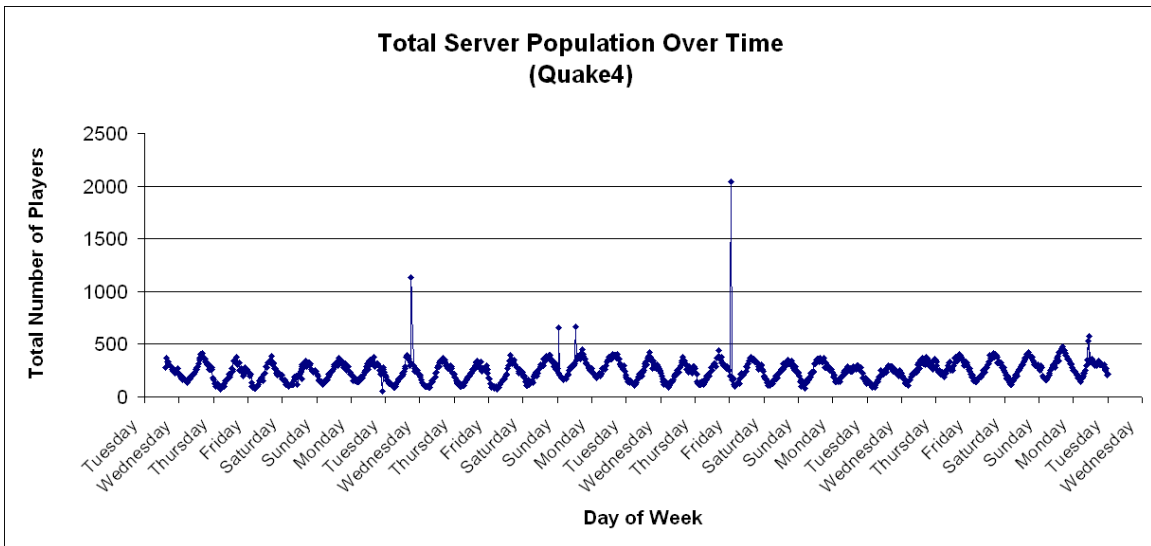


Figure 4-6

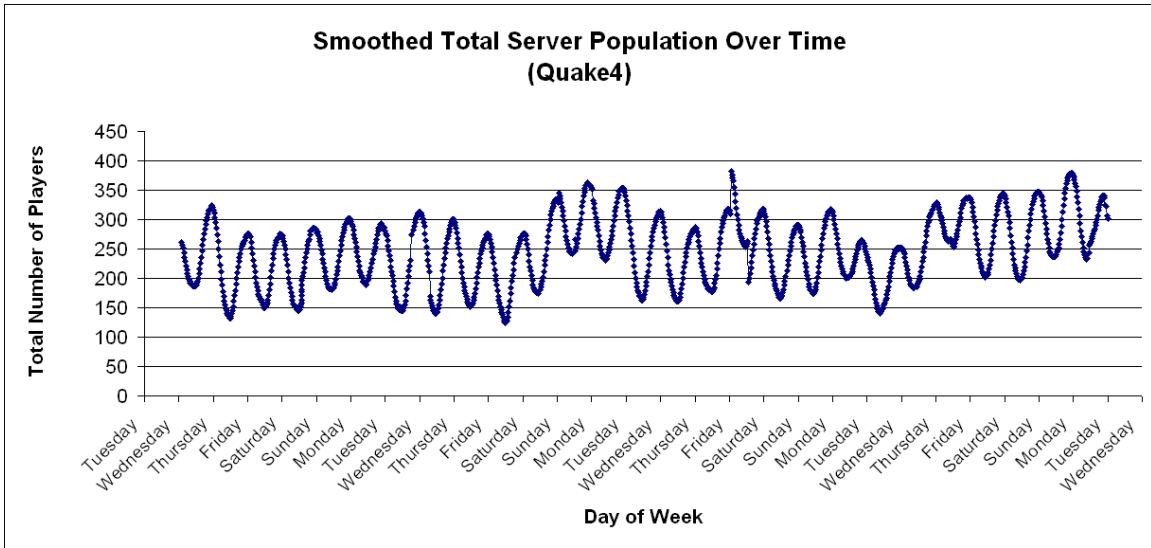


Figure 4-7

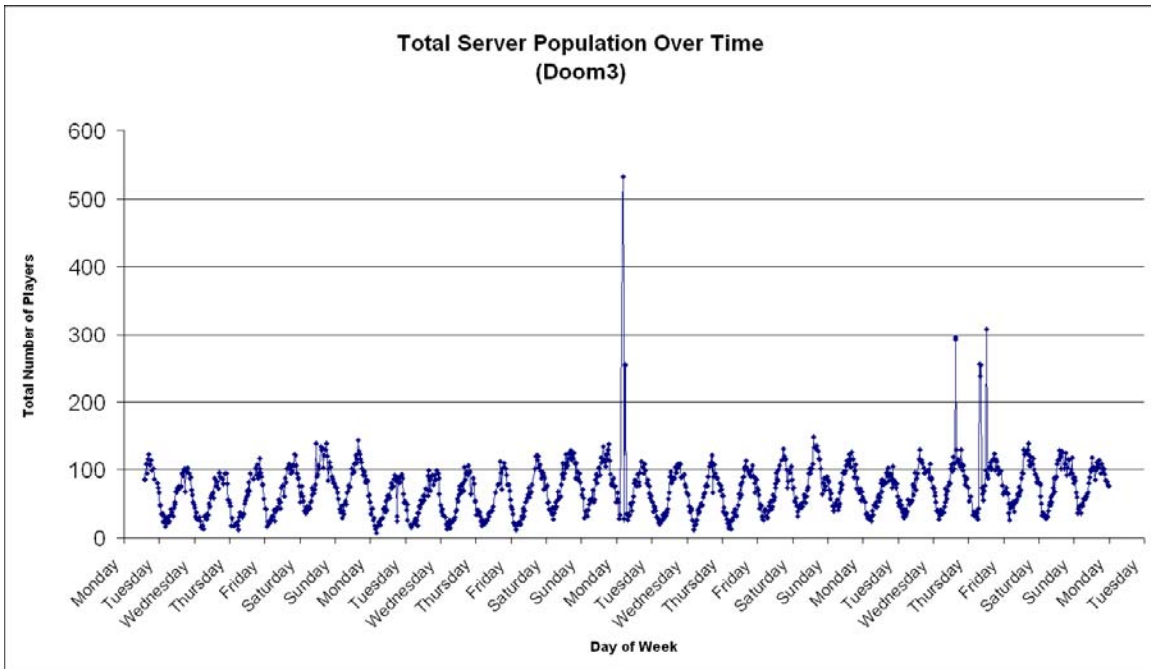


Figure 4-8

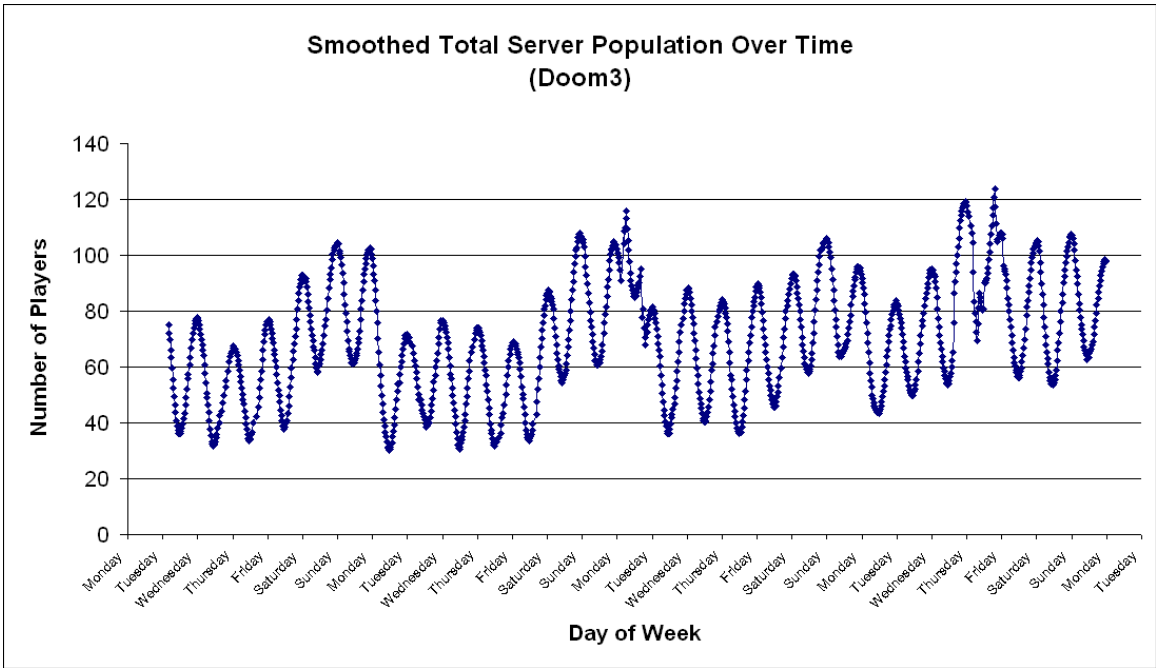


Figure 4-9

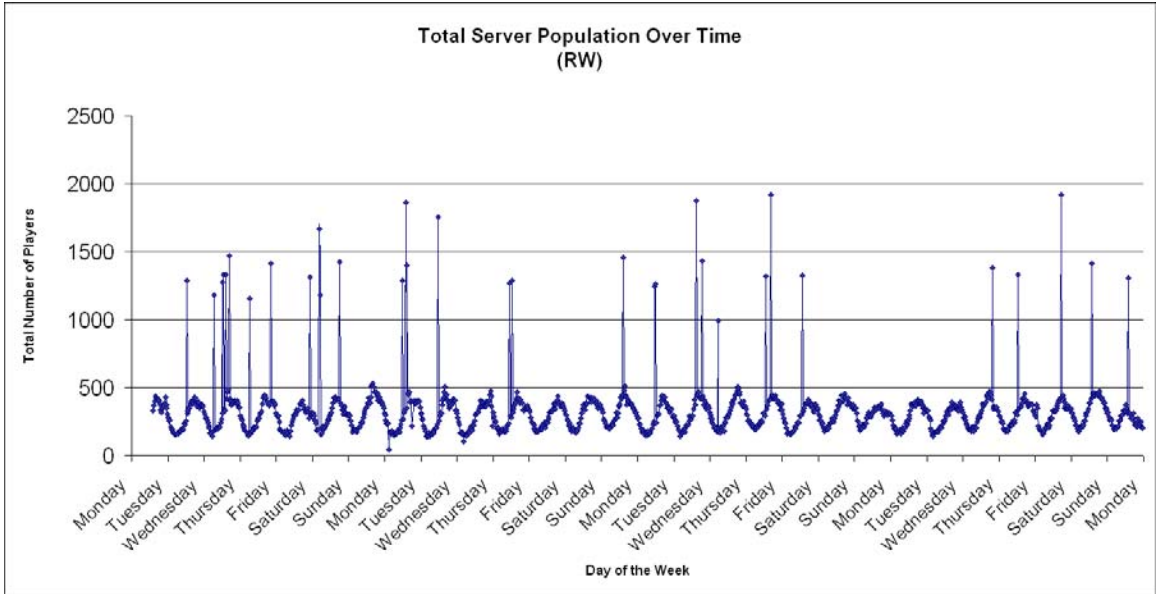


Figure 4-10

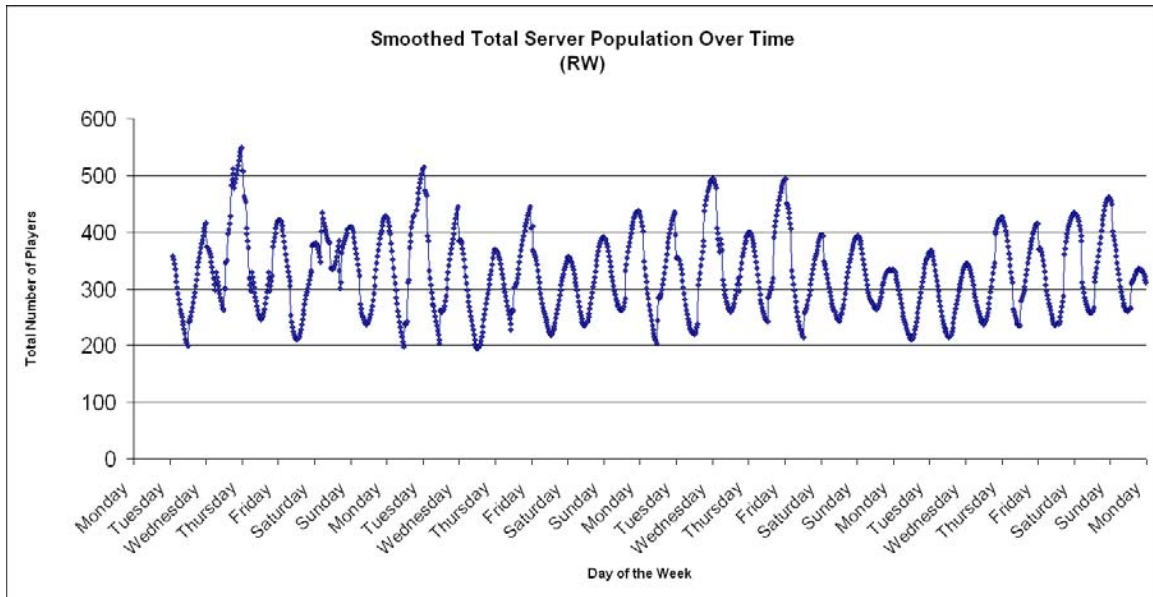


Figure 4-11

There is no visual difference in the total number of players based on the day of the week. Thus, all subsequent analysis is across all days. Choosing the “best” server then is mainly influenced then by the playability and fairness. The playability factor can be measured by the ping of the server when the client queries, as the server’s capability correlates to the server ping as demonstrated in the paper B. Jeremy, F. Safaei, and P. Boustead[9]. In this analysis we deal with the average ping because of the multiple clients, and since ping is a good indicator of playability we can infer that average ping can be a good indicator of the server’s capabilities for the group overall. The average ping, used throughout our analysis, is calculated across all clients for each query to a server. As for fairness, the standard deviation is a good indicator as shown in the paper by B. Jeremy, F. Safaei, and P. Boustead[10]. So we can apply this concept of standard deviation on our client to server pings to identify how fair it is for the individuals in the group.

Having identified both fairness and playability, it is important to understand that both are important measures for players in a group. Fairness is important to all players as individuals as they would want to be able to play at their best without having network or server problems that might impede them from doing so. For example, a high server ping indicates a high latency so the player will have a delay between their actions and the server results, which can affect them negatively in the game. Fairness is important overall to the group as the group would want to be able to play and enjoy the game together without having any members of the group be at a disadvantage compared to others due to outside reasons. One possible reason is that many games intentionally slow down or warp time to accommodate players with higher latency. For example, a group would not want members of the group to be hampered in the game because those players have a higher ping. Since these two characteristics have such an important role in the choosing of a server that would be considered “best” in a group’s standard, we believe that it is prudent to use them as a general heuristics to choose “best” server.

As pointed it out by [4], it has been proven that every game has a different ping threshold which when passed will affect the player’s ability to play. Because of this we expect that the games we examine to be no different, but since there are no studies indicating the threshold of these games we will be making assumptions on the threshold for these games. Using those thresholds we define how fair a game is depending on the maximum ping derived from the standard deviation calculations. The paper by M. Dick et al[4] that at certain intervals of ping the difference in playability of a game from a player’s perspective is not significant. For example in Unreal Tournament a player

having a ping of 25ms, a player with a ping of 75ms, and a player with 150ms will not notice much of a difference in the game.

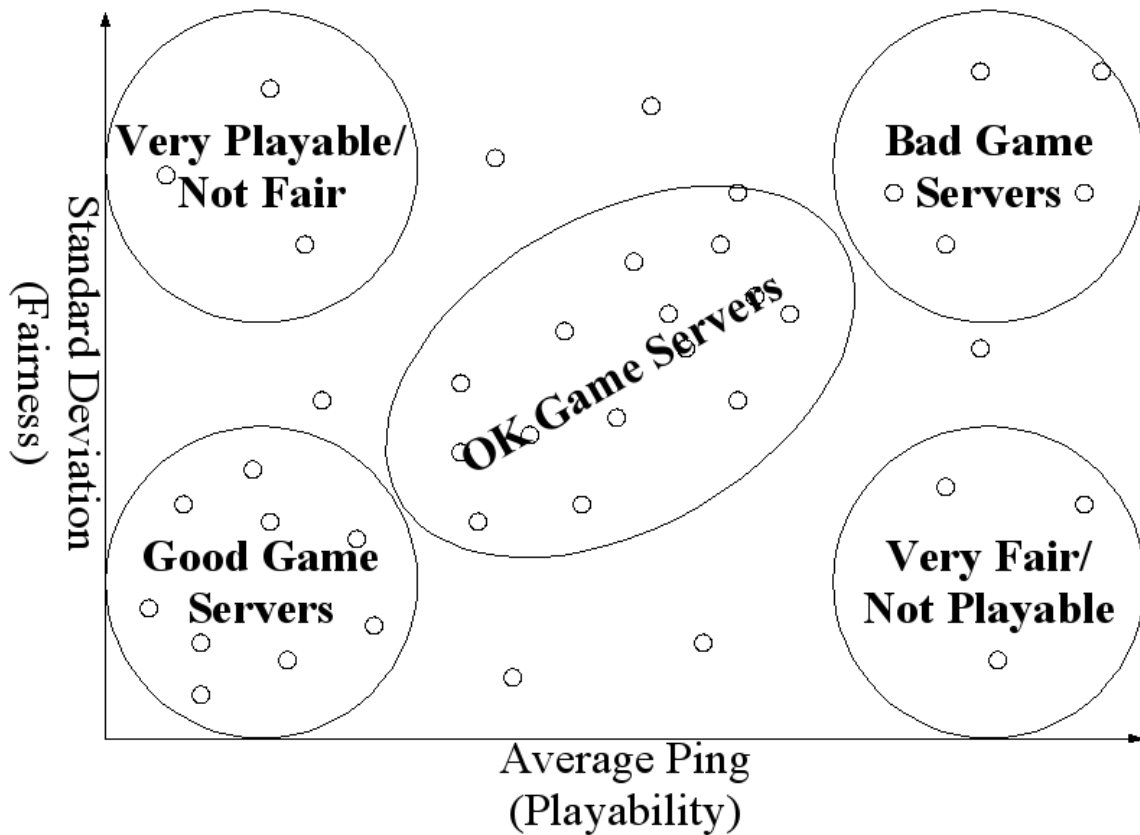


Figure 4-12

Standard deviation of the average ping is the metric we use to measure fairness as well as to calculate the maximum tolerance of ping for a server to be considered “good”, “okay”, or “bad”. For example, if a server has 100ms average ping and 200ms standard deviation then at one standard deviation would mean that the maximum tolerance of ping possible on that server would be 300ms and would mean that the server is “okay”. The limit on standard deviation is rather low as one standard deviation only encompasses 68% of players in the server while two standard deviations encompass 95%. So for a server with a standard deviation of 100ms, it only accounts for 68% of players in the server while raising it to 200ms would then encompass 95% of the players. Then if that server

has an average ping of 150ms that means that 95% of the players are between the range of 0 to 350ms putting it on the edge with some player's limits on acceptable. If the same is applied but with higher values like 200ms standard deviation and 300ms average ping then 95% of the server's population is between 0 and 700ms. Most players in FPS games consider anything above 500ms unplayable, as playing FPS games requires fast reaction times and having half a second of delay hampers the player's ability to react quickly.

Figure 4-12 is a visual comparison of standard deviation and average ping illustrated with made up data. Examining the graph, the lower left corner is the location where the "best" servers are the ones with the lowest average ping and lowest standard deviation. Meaning those servers are the fastest and fairest as they have the lowest ping overall for the players and the difference between the pings of the players are minimal. Looking at the graph, the worst servers are the ones directly opposite of the good servers, so the upper right corner where the average ping and standard deviation are highest. The servers that fall between the range of good and bad servers are those that are decent or okay for the group as they will have a varying mix of playability and fairness. Other servers that are to be avoided are servers along the top as they are unfair with their high standard deviation and the servers on the far right of the x-axis as they indicate low playability with their high ping.

Due to time constraints we were only able to do a full analysis on Doom3, Quake3, and Quake4. Examining the data gathered from 3/13/2007 to 3/19/2007 of the Doom3, Quake3, and Quake4 servers from different clients and plotting them we acquire the graphs on Figure 4-13, Figure 4-14, Figure 4-15, and Figure 4-16. Each of the points in the graphs represents one instance of one server across all clients during our queries.

Looking at the data gathered during this time period it can be observed that a fair number of servers have a good combination of both low ping average and standard deviation. We consider pings from 0ms to 200ms and standard deviation below 100ms to be “good”, between 200ms to 400ms and 100-300ms standard deviation to be “okay”, and from 400ms and 300ms standard deviation to be “bad”. All these values are derived based on [4] and intuition. From the values presented on UT2004 and Counter Strike it is not unreasonable to consider 300ms as a maximum ping to be “good” as from [4] it alludes to a ping of 150ms to not have much impact on player performance. Also mentioned in [4], that players underestimate the tolerance of ping in games, so a higher value than 150ms should still suffice as “good”. Also deriving from the findings in [4], where it states that certain games like Counter Strike are playable up to 500ms we considered it to be reasonable for the “okay” servers to tolerate up to a maximum of 100-300ms standard deviation along with the 200-400ms in average ping.

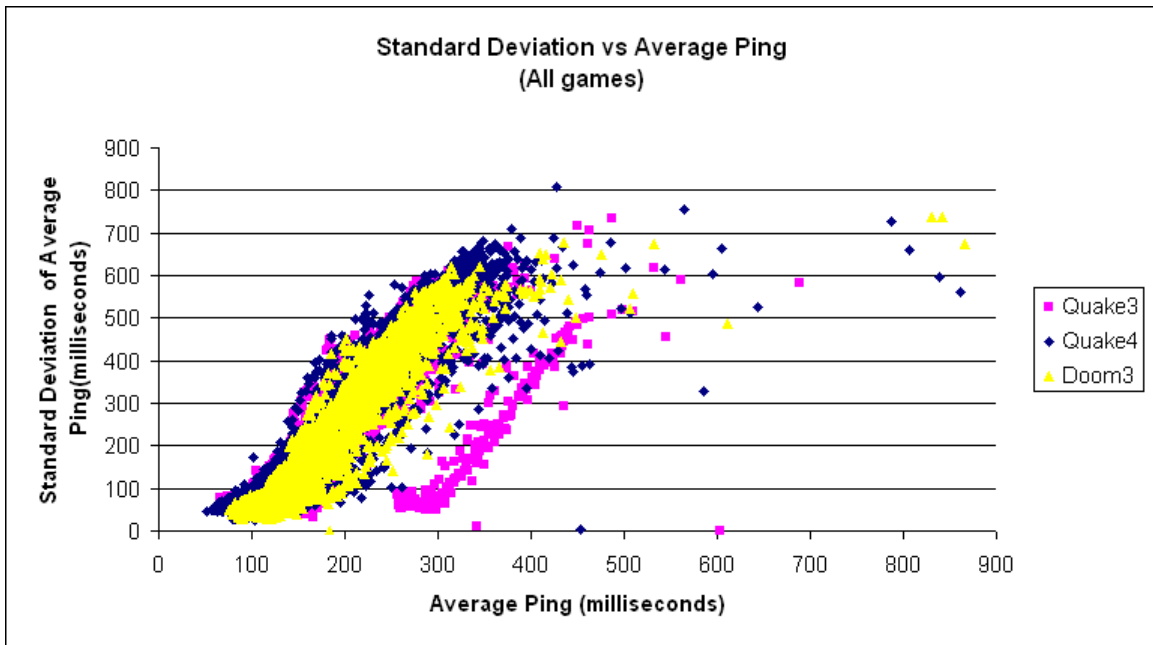


Figure 4-13

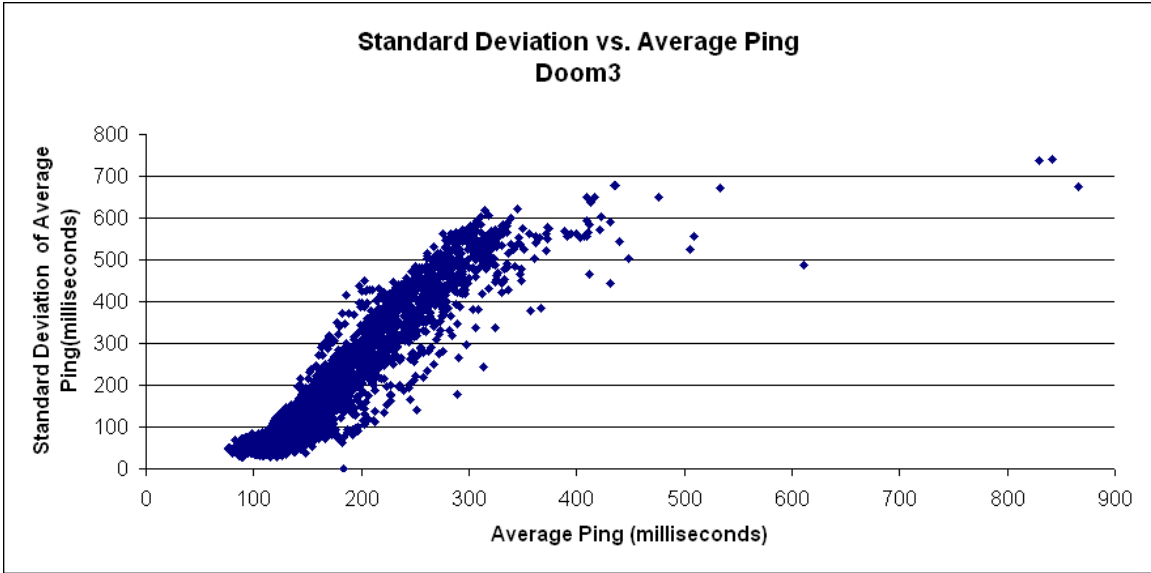


Figure 4-14

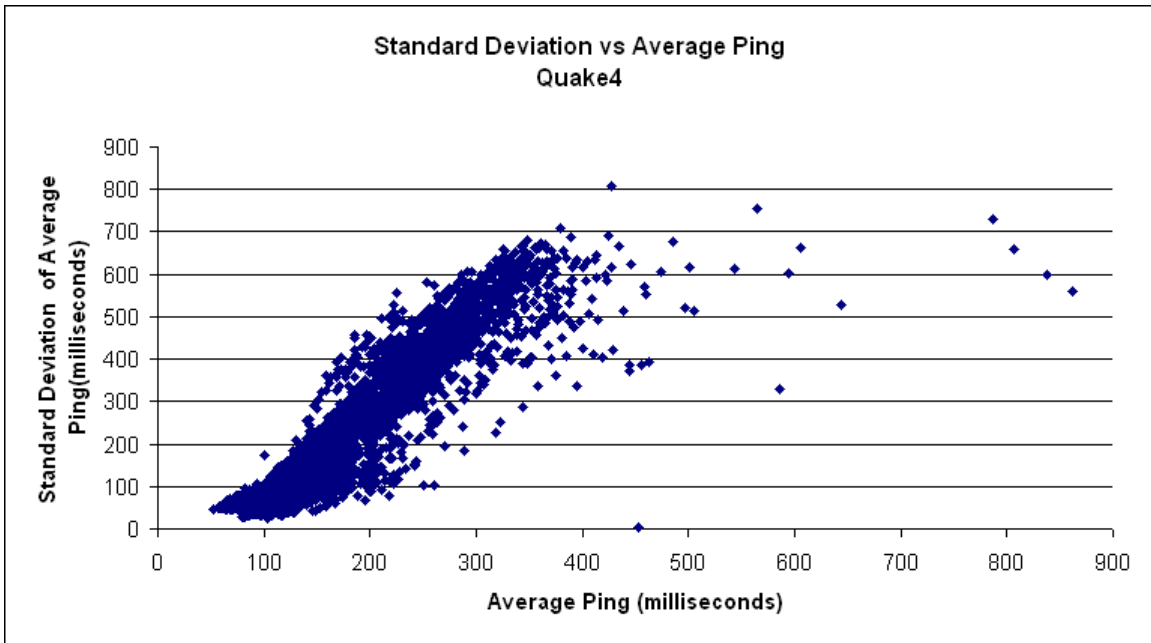


Figure 4-15

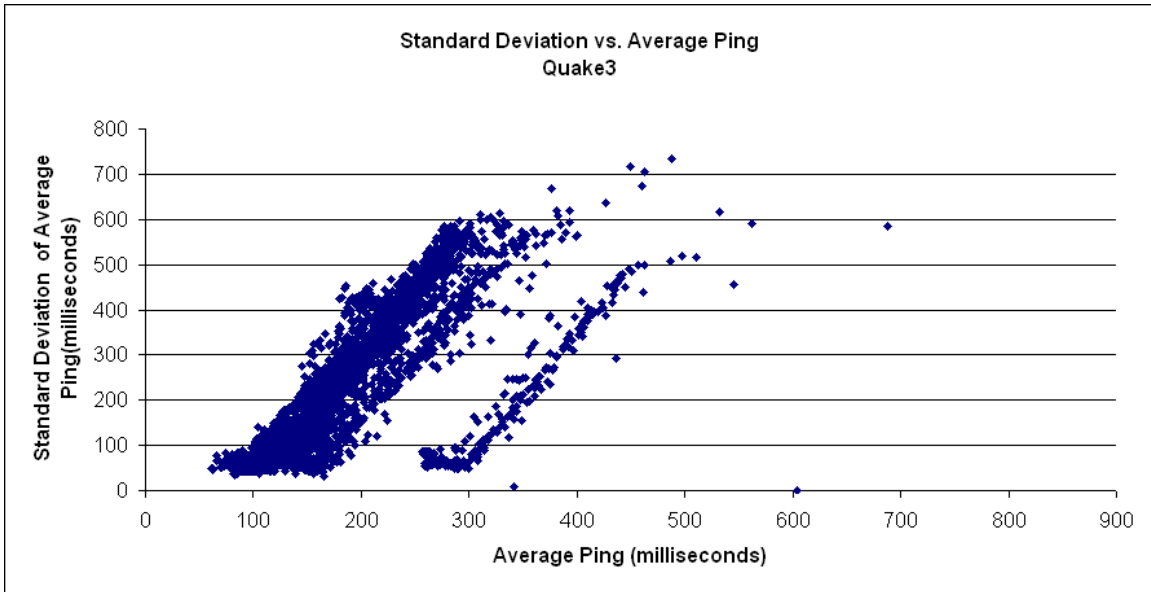


Figure 4-16

<u>All Games</u>	<i>Good</i>	<i>Okay</i>	<i>Bad</i>
Count	2856	4983	4653
Percent of Total	22.86%	39.89%	37.25%

Table 4-2

The range of servers that are okay is the largest as shown on *Table 4-2* for all games combined, since they do not excel in either average ping or standard deviation, as they have varying ranges of ping average and standard deviation. Surprisingly the number of servers that are unacceptable ping-wise and deviation-wise is rather large, with the number of servers that have values past the 400 ping and 300 standard deviation mark.

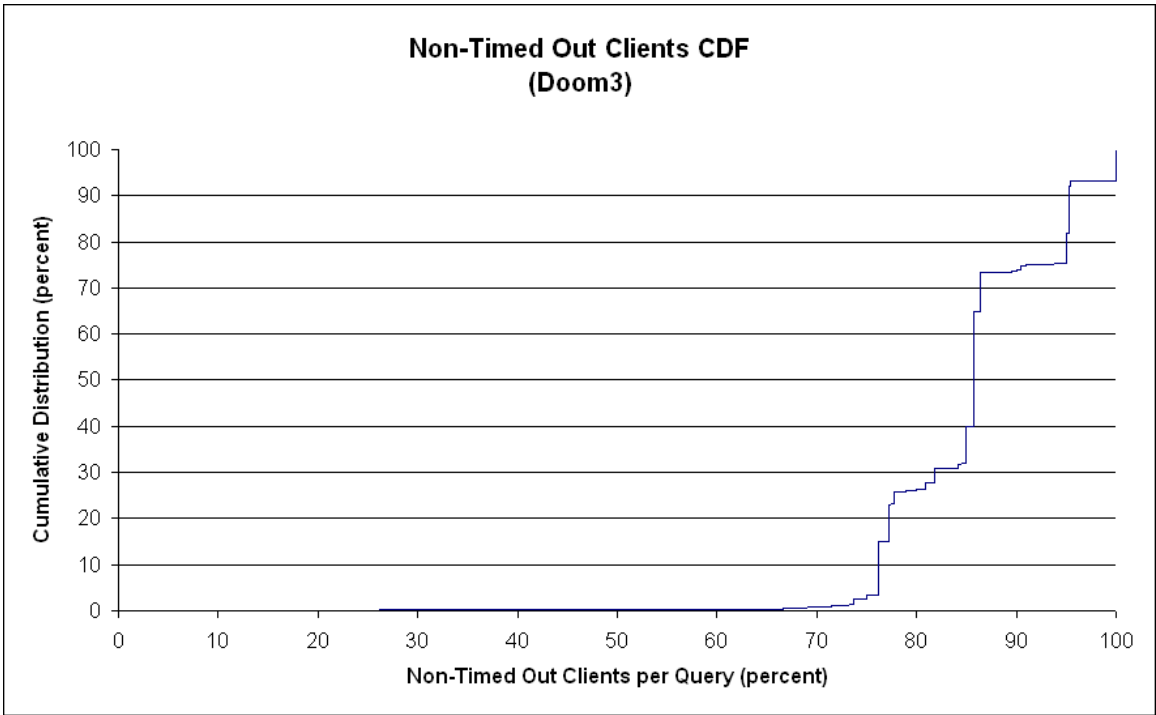


Figure 4-17

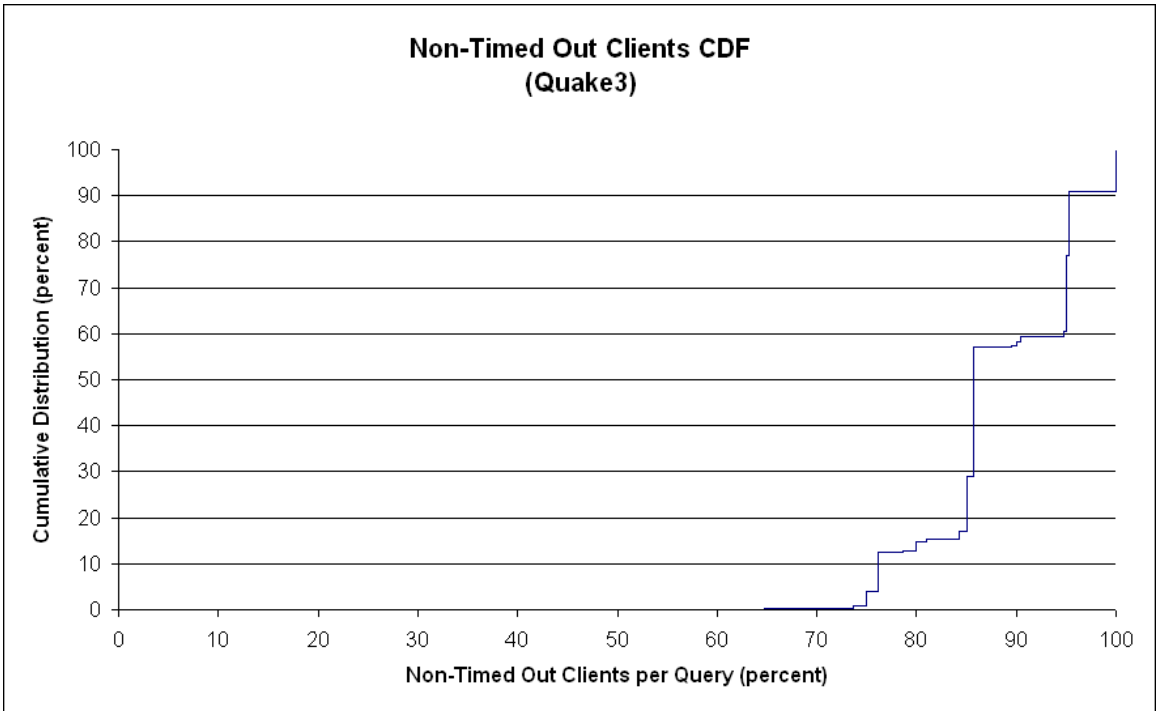


Figure 4-18

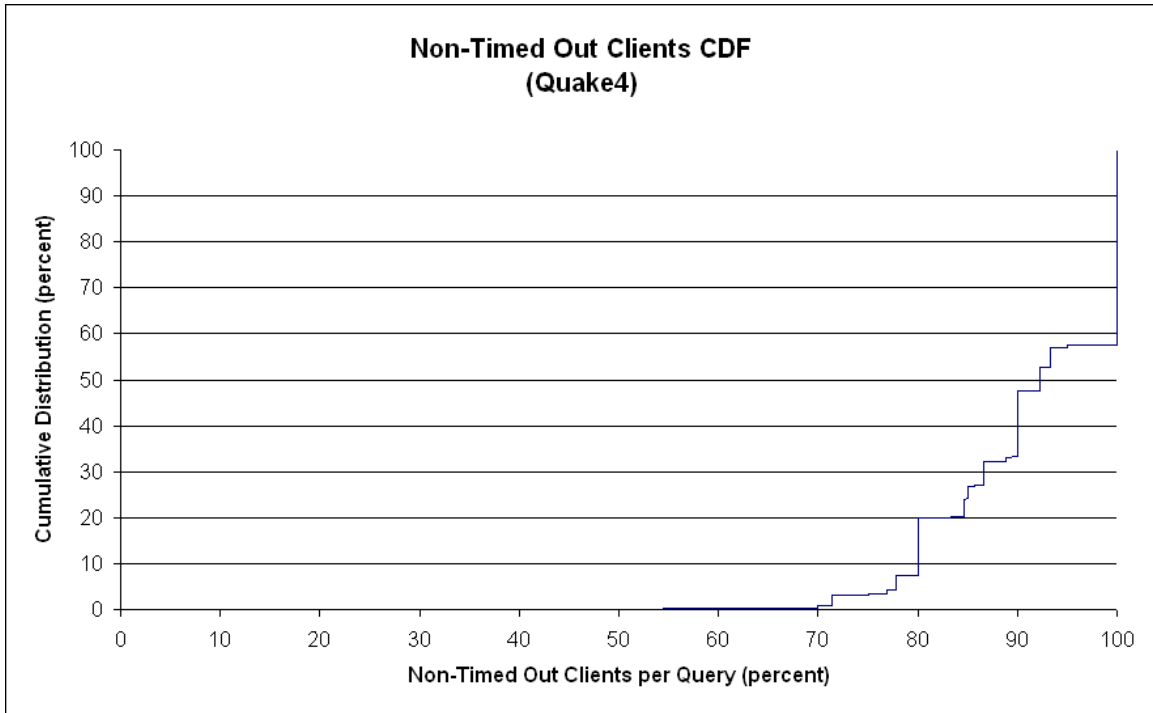


Figure 4-19

Looking at the graphs in *Figure 4-14*, *Figure 4-15*, and *Figure 4-16*, it can be noticed that there are some unusual data points. Most notably are the data points with a standard deviation of zero. These are a result of all of the clients querying a game server and all but one timing out, which gave us a single ping value and a standard deviation of zero. To see how often this occurs and to see how many of our data points are a result of only a few of our clients not timing out when querying a server, we calculated the percentage of our clients that didn't time out when query game servers. This data was then compiled into a cumulative distribution graph for Doom3, Quake3, and Quake4 shown in *Figure 4-17*, *Figure 4-18*, and *Figure 4-19*, respectively. What these show is that for even the first 1-3% the percentage of clients that returned ping values is 70% or better for all 3 games. This means there is a minimum of data points that result from only

a few of the clients returning pings while the others time out, and thus something we need not worry about for our data analysis.

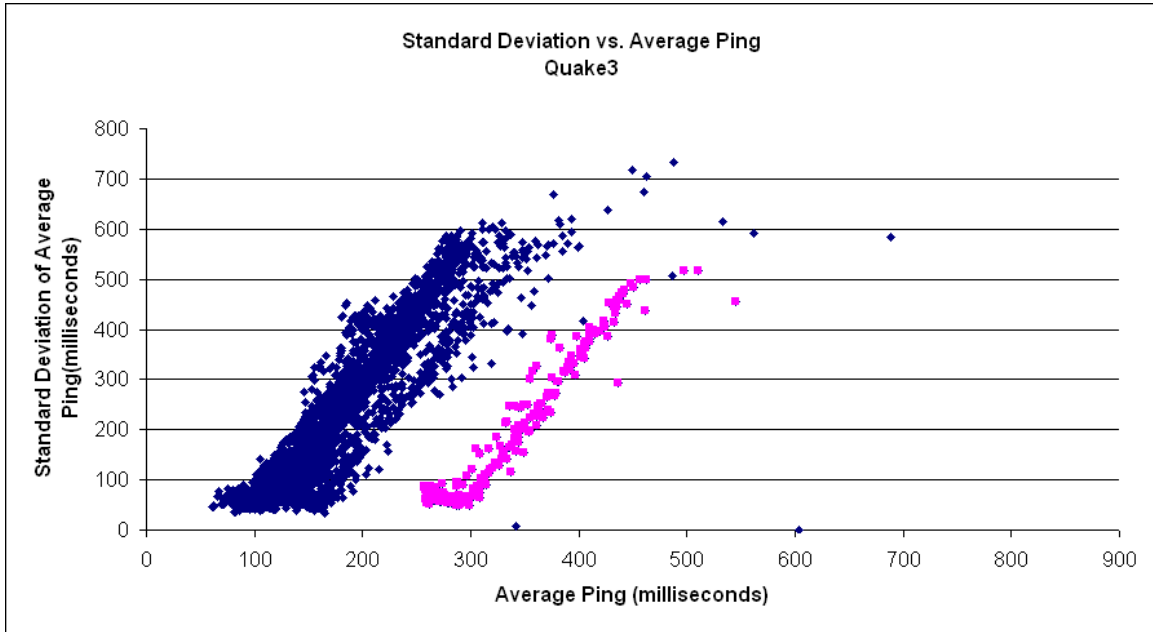


Figure 4-20

Looking at the Figure 4-14 we can see that the graph provided from Quake3 is very similar to that from Doom3 and Quake4 even though the games were made and published in different years. The main difference between the graphs is that Figure 4-16 has an anomaly on the right side of Figure 4-20, which can be traced down to the dataset from one specific server Quake3 set shown on Figure 4-21 below.

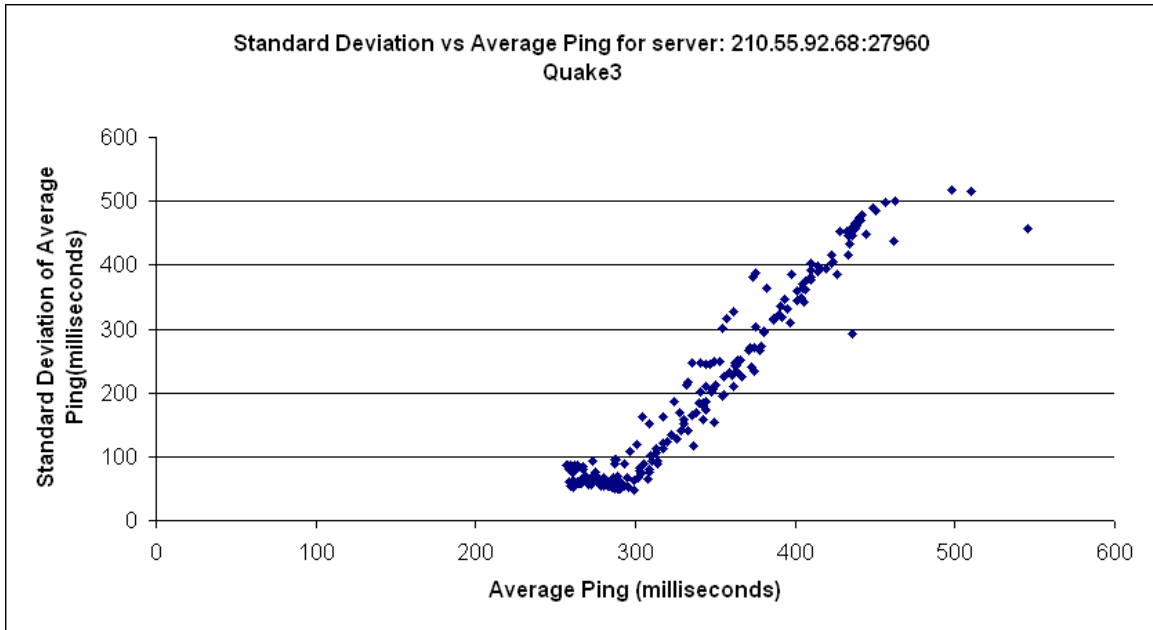


Figure 4-21

The data from the server that stands out from Figure 4-16 and Figure 4-20 however does not impact our conclusions about the threshold for determining “best” server. Instead it is an indication that while most servers for this game lie within a certain range for their average ping, this server does not perform as well though it is still an okay server when its instance is in the lower ping and standard deviation ranges. Applying the thresholds we have derived on this dataset we can observe the servers presented from Quake3, with the count of servers shown on Table 4-3 below, it will guarantee that a suitable server will be found as more than half of all the servers are “okay” or better. With the good servers from ranges 0 to 200ms average ping with up to 100ms deviation, okay servers with 200 to 400ms average ping with up to 300ms deviation, and bad servers with anything above 400ms average ping and more than 300ms deviation.

<u>Quake3</u>	Good	Okay	Bad
Count	170	2532	2513
Percent of Quake3 Total	3.26%	48.55%	48.19%

Table 4-3

Another difference in the graphs in *Figure 4-14*, and *Figure 4-15* is that the *Figure 4-14* which represents the data from Quake4, when compared to that of Quake3 there is a difference of about 100ms in both average ping and standard deviation. Most of the values for Quake3 are below 300ms average ping and 600ms standard deviation if the anomaly is ignored. Compared to that the majority of Quake4 servers have up to 400ms average ping and 700ms standard deviation. While the disparity is not very big it is interesting to note. This disparity could be possibly caused by a difference in protocols or due to a more complex game requiring more network resources. However when examining the data closer we can see through *Table 4-4* that Quake4 actually has a lot more “good” servers.

<u>Quake4</u>	Good	Okay	Bad
Count	1272	1454	1297
Percent of Quake4 Total	31.62%	36.14%	32.24%

Table 4-4

It is possible that this could be because while the game is newer and uses more resources which causes the 100ms difference but it could be that the players that a

majority of the players that play Quake4 have better systems and internet connections. This could indicate why there is more variance in standard deviation and average ping for Quake4 but still have a much bigger number of “good” servers than that for the game Quake3 as observed through *Table 4-3* and *Table 4-4*.

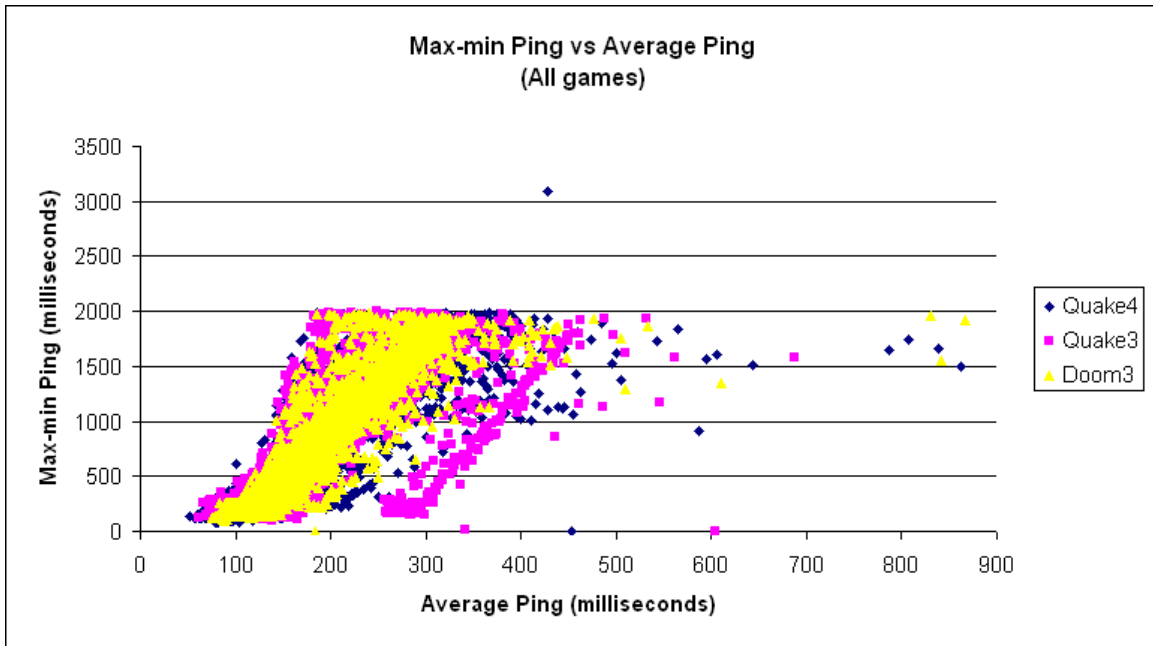


Figure 4-22

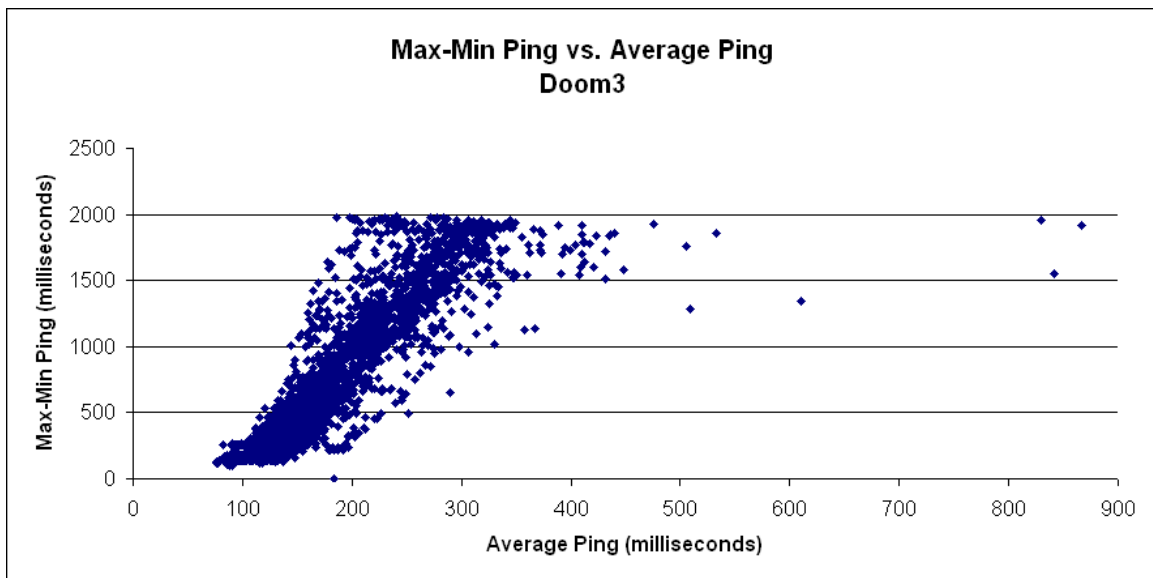


Figure 4-23

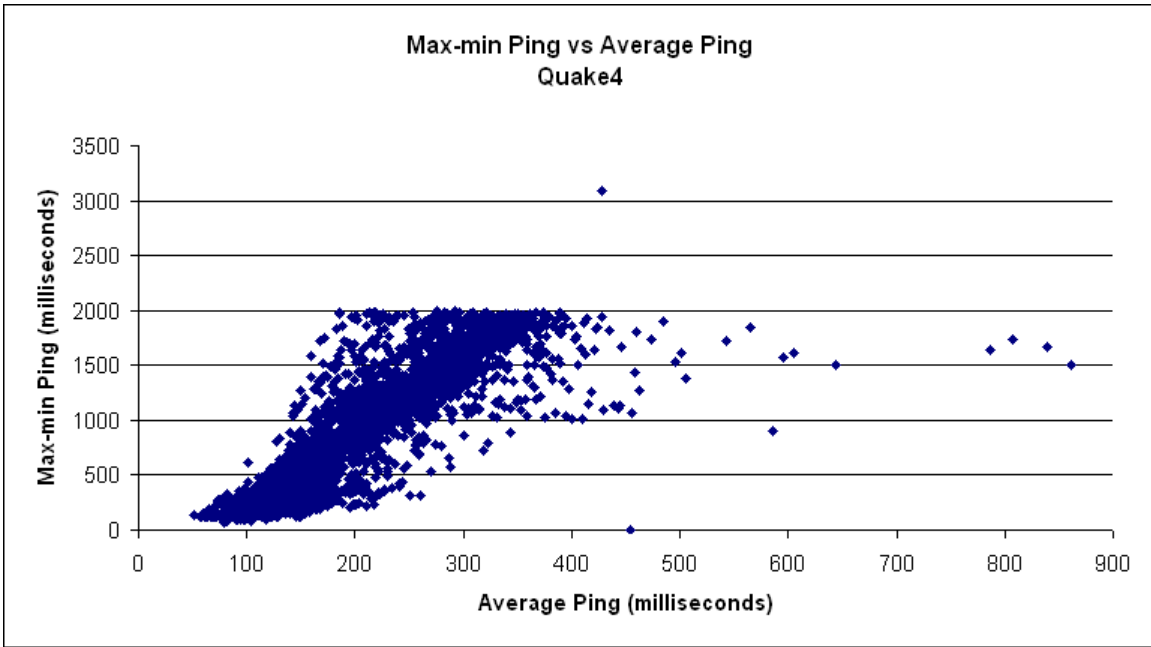


Figure 4-24

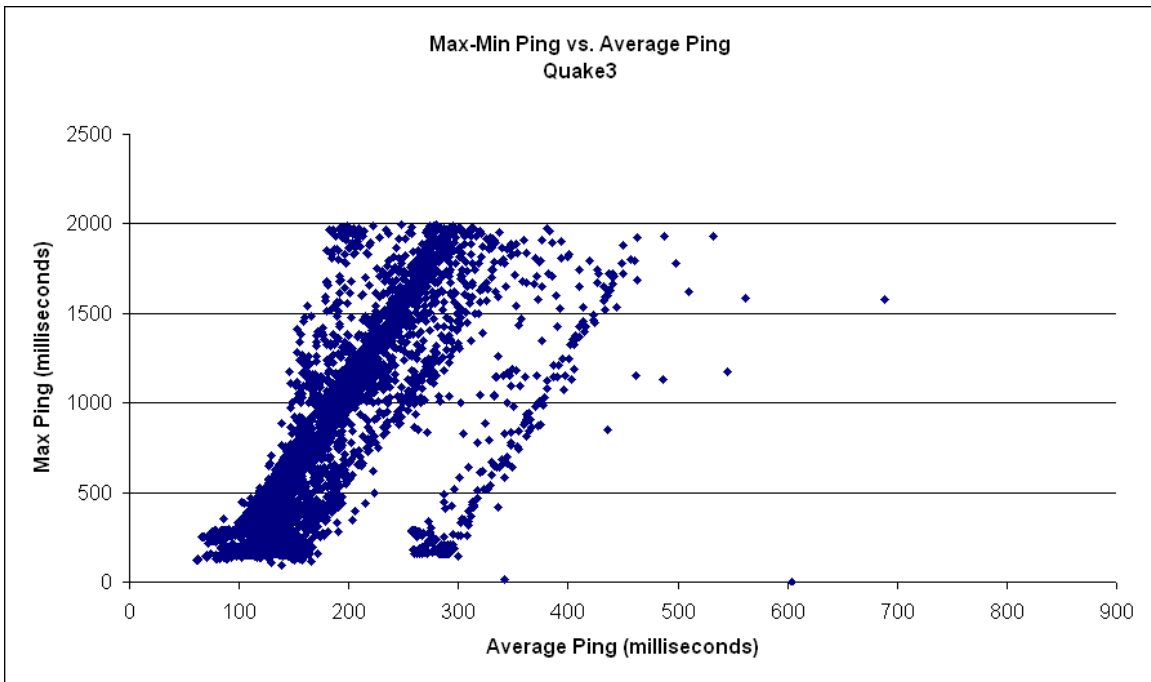


Figure 4-25

Another point of view of the data is shown in the Figure 4-23, Figure 4-24, and Figure 4-25 shown above and is comprised by the calculation of the maximum ping minus the minimum ping and the average ping. The graphs show more conclusive data specifically on the maximum variance of ping found in the servers queried rather than the

implied maximum values presented by the previous graph in Figure 4-14, Figure 4-15, and Figure 4-16. Since all the graphs, Figure 4-14 through Figure 4-16, represent the variance in ping values of servers from the client point of view but in different methods, they look similar to each other. This graph further supports the boundaries we derived for “good”, “okay”, and “bad” servers based on playability and fairness, in the context of Doom3 servers, as it shows more concretely the reason behind choosing such combinations of low values for average ping and standard deviation.

When we compare the standard deviation vs. ping graph with the maximum minus minimum ping graph we get very similar results as shown on Figure 4-23, Figure 4-24, and Figure 4-25 above. The disparities from the graphs can be constituted by some of the maximum values being lower than the maximum value encompassed by the deviation or the maximum value being slightly higher than the deviation. These disparities however do not conflict with our previous observations, as there are only a few and more often than not are outliers. We also compared the standard deviation vs. ping graphs with the maximum ping vs. average ping graphs, but these graphs ended up looking very similar to the maximum minus minimum ping vs. average ping graphs and are just shown in Appendix A1.

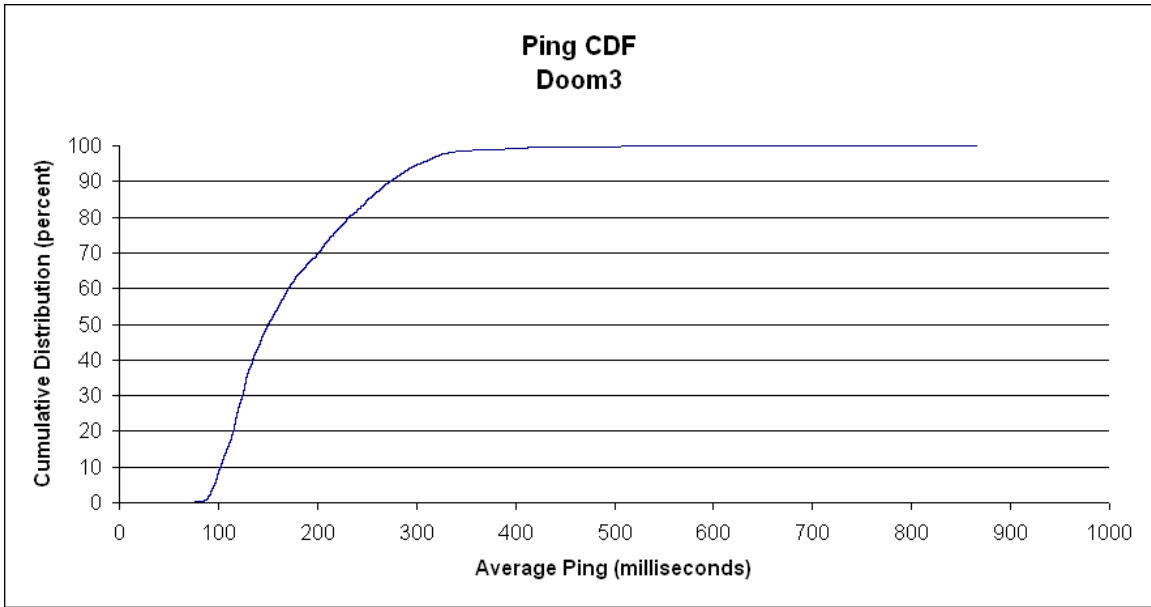


Figure 4-26

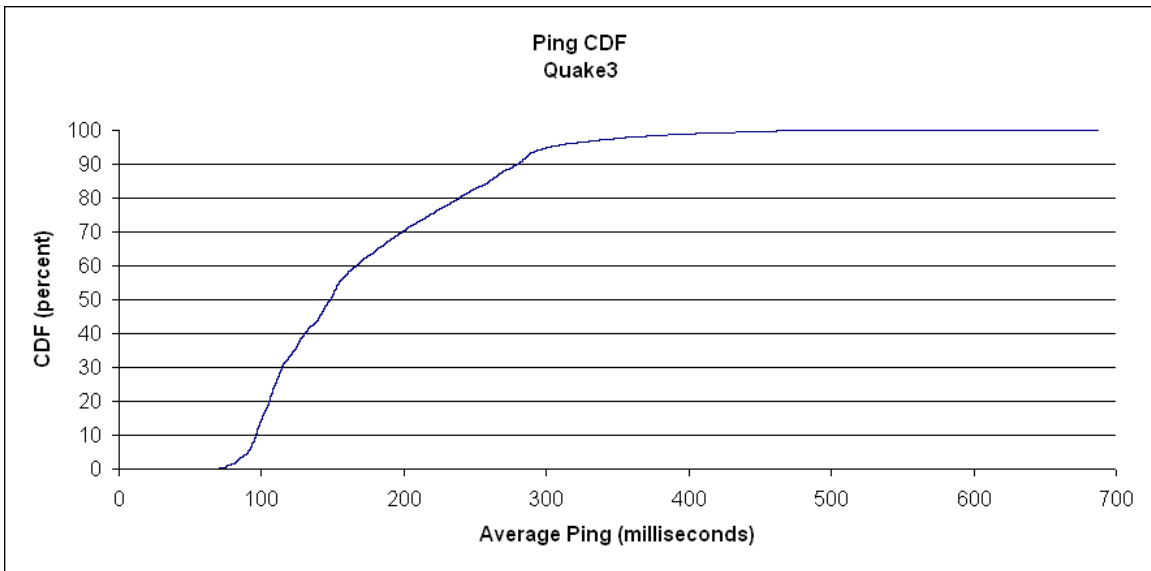


Figure 4-27

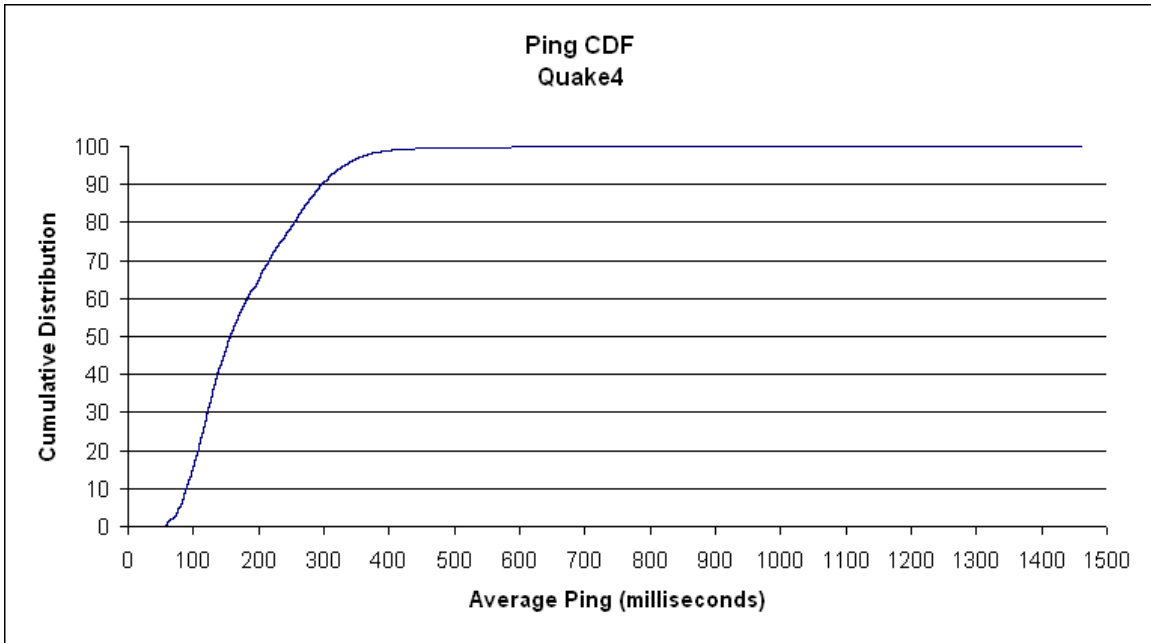


Figure 4-28

Figure 4-26, Figure 4-27, and Figure 4-28 are Cumulative Distribution graphs of average pings for all servers for all three games.

Further ping analysis through the use of these cumulative distribution graphs, we can observe that over 90% of the queries from all the different clients have an average ping of 300ms or less. This indicates that our demands for the limits are suitable and plausible as the ping of players are within the ranges of our limits. Looking at the values present, 60-70% of the queries from all the different clients were 200ms or less on average making it plausible for said clients to find a server within the good category and guarantees a server that is okay for a group of people.

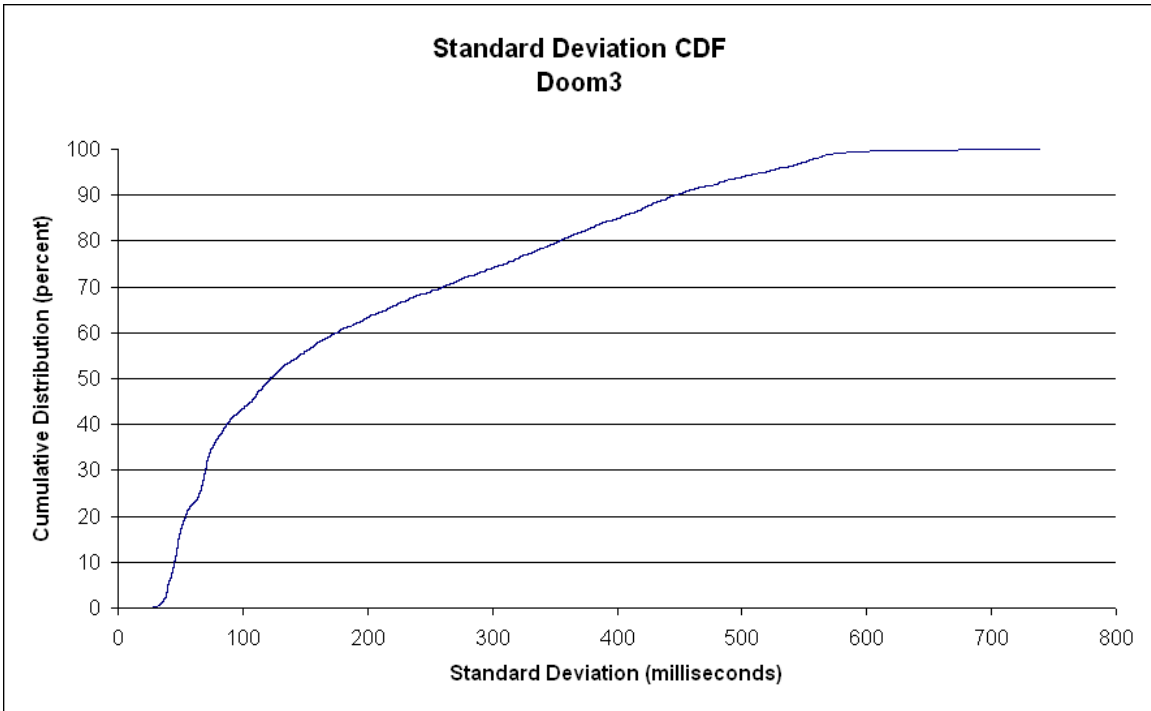


Figure 4-29

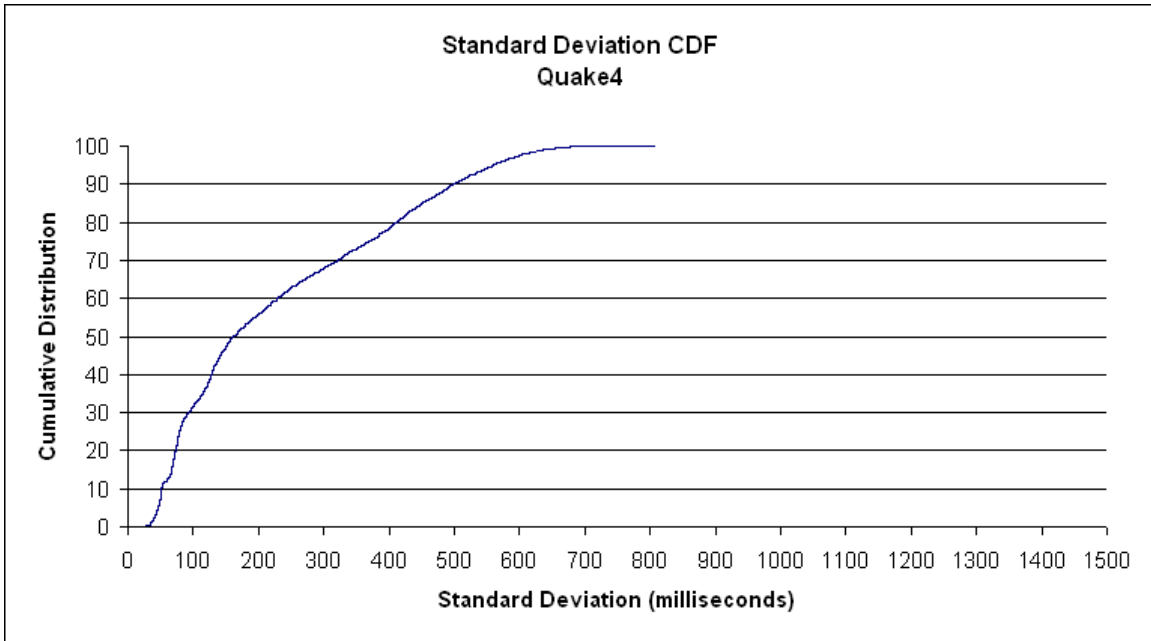


Figure 4-30

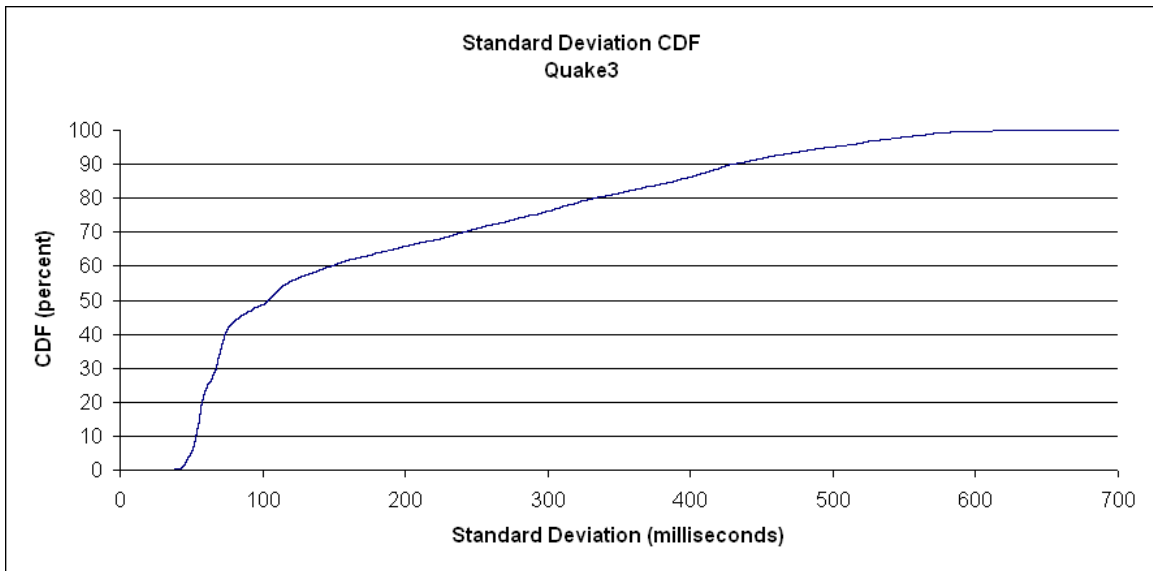


Figure 4-31

The Cumulative Distribution graphs on standard deviation for all three games show that despite the good average pings that are within each query, the variance of pings can be rather large. This just highlights that the playability metric, the average ping, by itself is not good enough to use to choose a “best” server when in a group context. However, the data shown in Figure 4-29, Figure 4-30, and Figure 4-31 still indicates that our fairness and playability metric provides a plausible way of attaining a “best” server. While the graphs indicate that normally only 40-50% of the queries’ standard deviation was 100ms or less, it does not indicate that a “good” server cannot be found. It just indicates it is not always possible for this method to find a “good” server, but it does not stop it from finding an “okay” server. While players might be only looking for “good” servers and it might seem that an “okay” server is not good enough, it might be that the “okay” server is the “best” server available for their group. Since the okay server threshold is at 300ms deviation, by examining the graphs we can observe that more than 70% of the queries have pings within that deviation.

However, upon closer inspection the base location of the group is very important. For example it is easier to find a suitable server for a group of players who are all located within North America rather than a group that has players located in North America, Australia, and Europe.

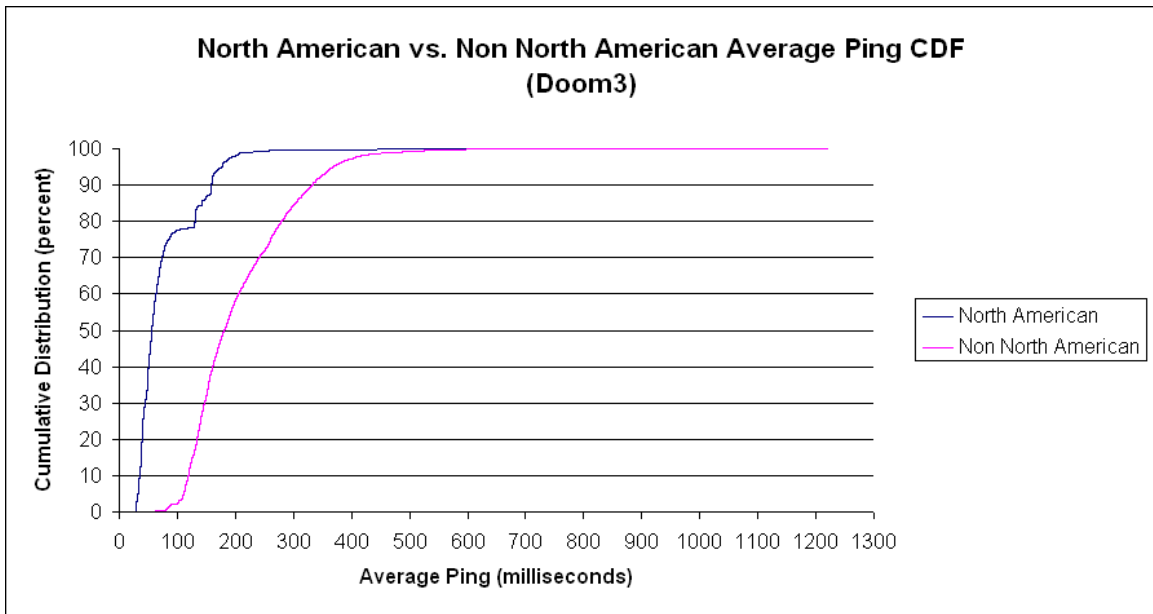


Figure 4-32

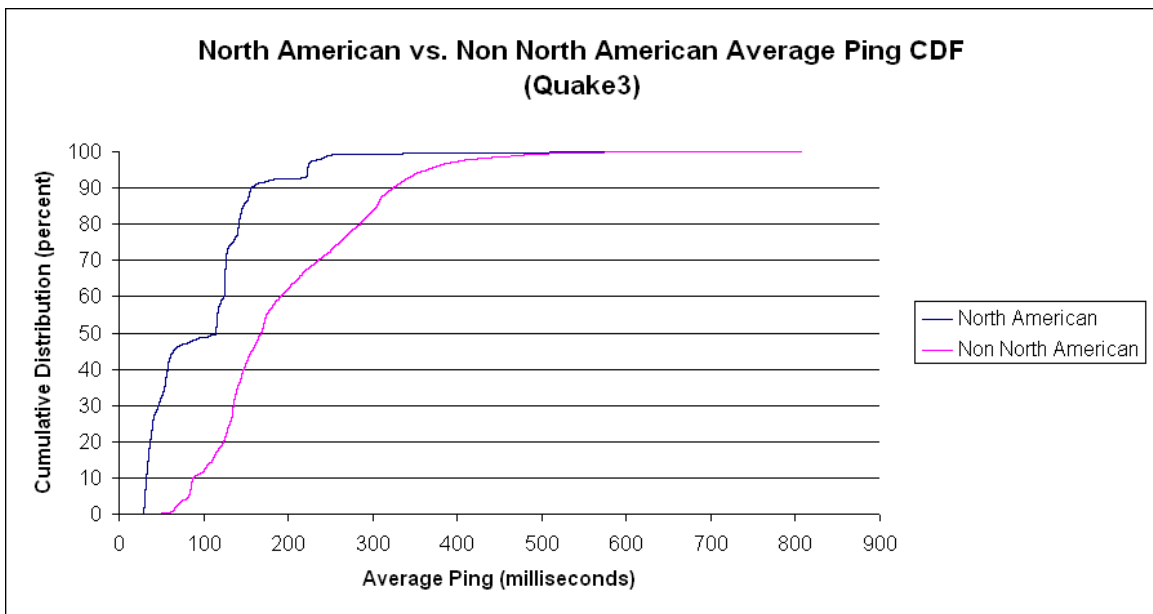


Figure 4-33

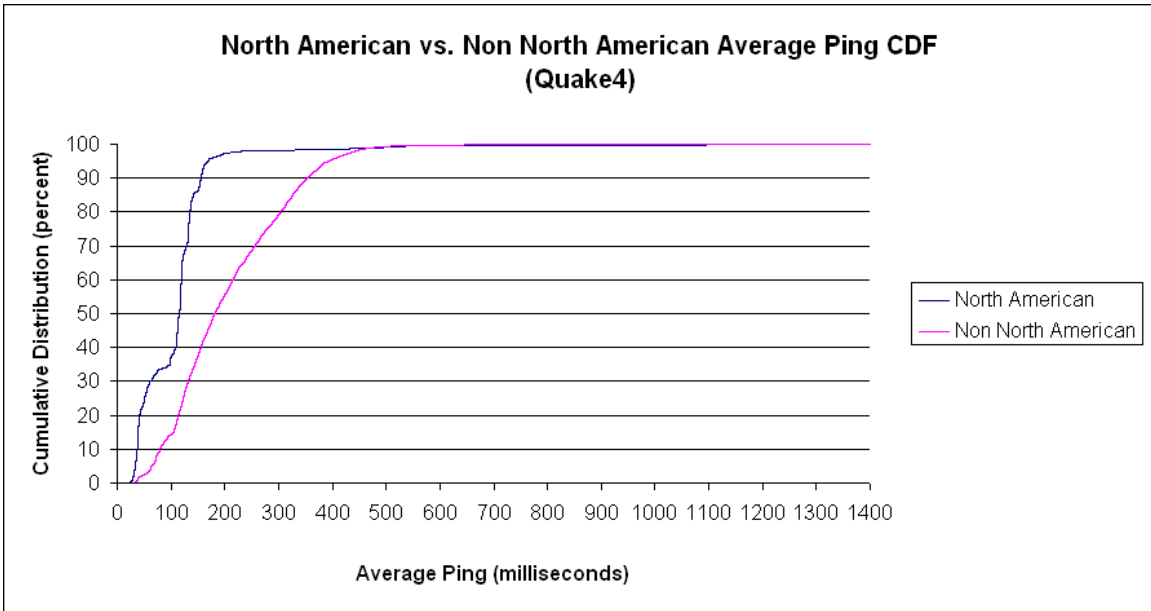


Figure 4-34

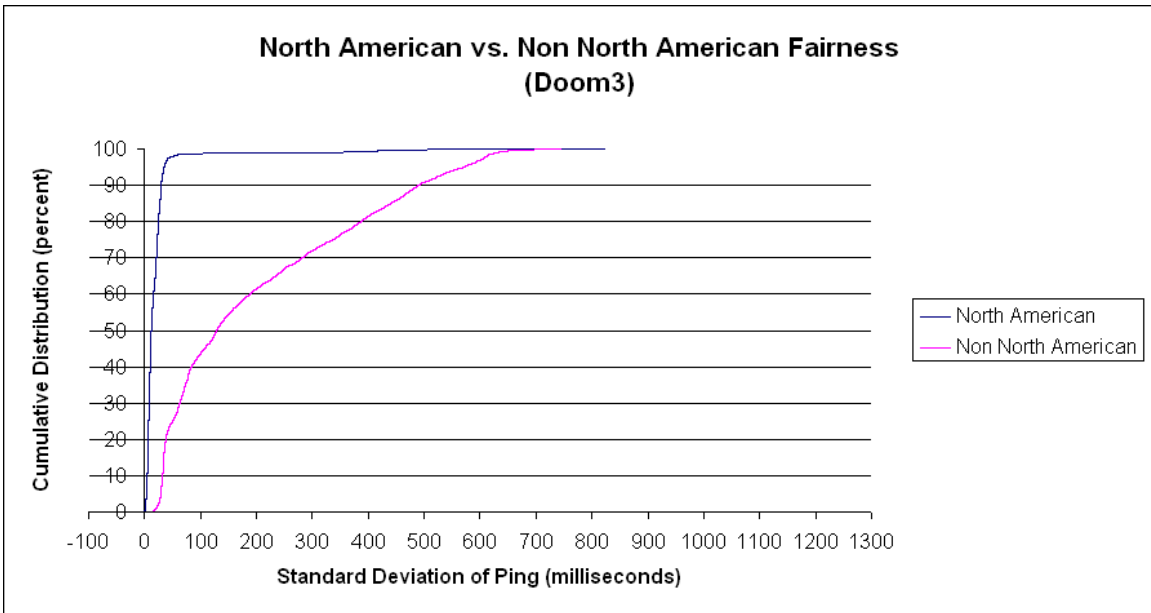


Figure 4-35

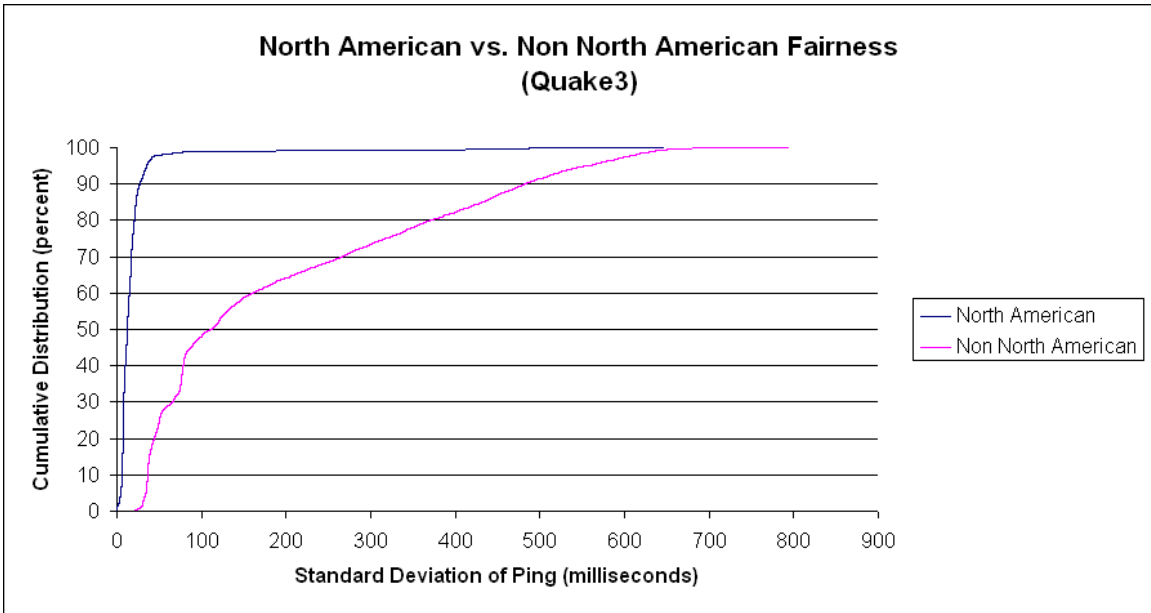


Figure 4-36

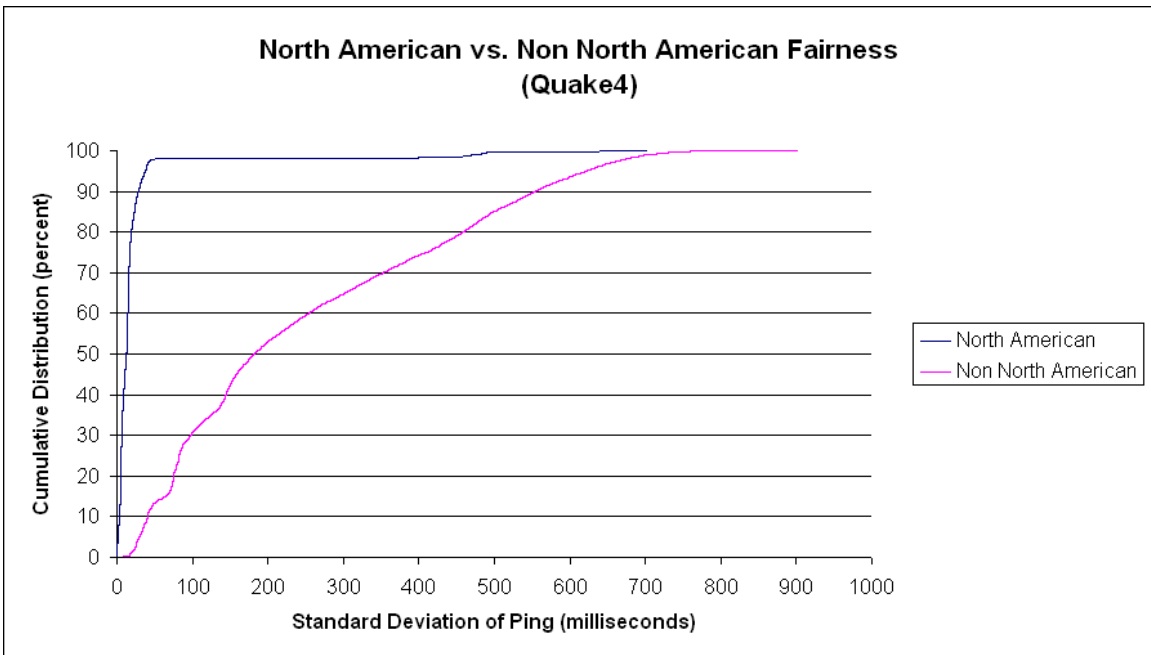


Figure 4-37

As shown in Figure 4-32, Figure 4-33, and Figure 4-34 about 90% of the North American clients have an average ping of 150ms or less while only 60% or less of the Non North American clients have 200ms or less average ping across all three games examined. While a few of the Non North American clients do have comparable pings,

the disparity in average ping between different regions is made apparent. This disparity makes it quite hard to find playable servers for the individuals in the group. Also looking at *Figure 4-35*, *Figure 4-36*, and *Figure 4-37* it shows that 90% of the North American clients have a standard deviation of 100ms or less while only 60-75% of the Non North American clients have a standard deviation of 300ms or less. Between playability and fairness it is obvious that fairness seems to be the biggest problem when searching for a suitable server. This is because the standard deviation between the players from different regions is the biggest disparity. At 90% of the Non North American clients the standard deviation is 500ms which for most players it is considered unplayable, while 90% of the Non North American clients have 350ms or less in average ping. With that big of a variance it is more difficult to find a server that will fair to all players.

4.4 Feasibility

This heuristic for determining a “best” server for a group of players while plausible to implement and work on a server browser it does have limitations. The limitations however are more of general limitations on any heuristics used to search for a server. The limitations are:

- Number of maximum players on a server
- Maximum number of servers to query
- How far players are from each other

The number of maximum players on a server is a limitation to the heuristic because depending on how many players the group is compromised of it will be harder to

find a suitable server. If there is a group of 24 players, it will be hard to determine how good a server is as most servers will not be able to accommodate that many players unless it is empty. However, it is hard to determine how suitable empty servers are since their ping is not representative of how the server will perform when it has population. Also if the group is large enough, 32 or larger it might be impossible to find a server as there might not be a server that supports that many players that is not already full or be partially filled with players.

Another limitation is the maximum number of servers to query since it takes time to query servers. While the amount of time needed to query a server is not very large, when querying a thousand or more servers possibly, the total cumulative time of the queries can add up. Also, if the server browser were implemented by a third party to the company that produced the game, it would not necessarily be done with full knowledge on the mechanics for querying servers for that game. This would make the queries less efficient and at worst, not work at times. This could be alleviated if the developer of a given game were to implement a group server browser themselves, since they would have all the information necessary to make the process as efficient as possible.

The distance between the players can be an issue too. This is because if players are very far apart then there might not be a server that will be both playable and fair to all the players. An example of this is Australian and North American players trying to play together. Any server within North America will be likely to have a ping of over 1 second to any Australian player and vice versa for Australian servers. The solution would be to find a server between the countries but due to geographical limitations it is not possible,

for example the Pacific Ocean. Choosing an Asian server does not solve the problem as it will most likely have a ping of over 1 second still.

5 Conclusion

Playing FPS games as a group, presents itself as a unique problem. Current server browsers do not provide all the tools necessary to effectively pick a server that is suitable for all people wanting to play together. As a result we explored a technique for determining a suitable set of servers for a group of players. This method involved using the average ping among players as a measure of playability and the standard deviation of the average as a measure of fairness. Using these two metrics we were able to define boundaries between “good”, “okay”, and “bad” servers. To make the problem more manageable, we assumed that players in a group look for the lowest ping, a certain number of players in the server depending on group size, and that all the players can get into the servers.

We selected the game Doom 3, Quake 3, and Quake 4 for the data gathering and the analysis. To actually get the data we used a command line multi-game server browser called Qstat and a collection of machines distributed over the globe called Planet Lab. We used Qstat to make queries to the game servers of each of the games, which was handled by our own software that was written to gather and organize the data generated by the Qstat queries. Our software was then put on a set of Planet Lab nodes and executed simultaneously using another piece of software we wrote that organized all of the queries the Planet Lab nodes were performing.

To get a better understanding of the game servers that we would be gathered data on, we first gathered data on all the servers listed by a game's master server. Based on the large number and variety of the servers, and the fact that many of the servers can be susceptible to down time, which would only lengthen the time to query a set of servers, we decided to narrow the list of servers to query down to a set list. In addition, we were able to determine that the day of the week has little affect on the data we would be gathering, and thus not a factor to be considered when doing our analysis. To generate our static list of game servers to query for each, we looked at how often the servers were up and how populated they were. We used this data to select 20 servers from each game to make individual Qstat queries on.

Once we had gathered the data on the game servers we had chosen, we applied our metrics of fairness and playability. We then plotted this data in order to define the boundaries between "good", "ok", and "bad" servers, defined based off of earlier work. Looking at the plotted data and how the points fit within the boundaries we defined; we found that server performance appeared approximately the same. This meant that regardless of the game, this method of choosing the best game servers works.

To further confirm that our technique worked, we looked at the cumulative distribution of the average ping and standard deviation of each game. The cumulative distributions of the average ping showed us that for 90% of the queries, the average ping was under 300ms and even more importantly that 60-70% of the average pings were under 200ms. This meant that on average there was a good chance of finding a "good" server for a group, and almost guaranteed that at least an "okay" server could be found. Looking at the cumulative distribution graphs for the standard deviation, we found that

while most average pings were well within the “good” server range, the pings that make up the average can vary a lot. With only 40-50% of the queries being in the “good” range, it would seem that this method may not be effective. However, this only showed that playability is not the only measure needed for finding the “best” server. Taking playability into account, a “good” server might be less likely to find, but an “okay” can still be found, with 70% of the queries being below the 300ms standard deviation mark for “okay” servers.

At the end of all this, we found that using fairness and playability for determining the best game servers was an effective method. These two metrics were good at representing the factors that are most important to players looking for a game server to play on. Finding “good” servers was entirely possible with the limits we choose even with players spread out over the world. Even more important was that finding an “okay” server is just about guaranteed. This means that a group server browser could be successively designed around using fairness and playability for determining the best servers for any group of players.

6 Future Work

Since not everything that we thought of to do for this project could be done given the constraints on time and resources, we present further work that can be done using some subset of the tools and methods we developed.

While we assumed that the group server browser would be capable of producing query information quickly enough to not have network or server conditions change significantly, future work could gather data on the volatility of network conditions. Part of determining whether this is possible would be to collect data on how long each client takes to perform a single or set of queries. Then using this timing data look at what the worst case is, then determine approximately how much network and server conditions change over the time span based on the worst case. This would require a client that can quickly query any of the game servers to ensure data gathered before and after the time span is accurate.

A simple extension of this work would be to simply perform the same data gathering and analysis for more games and more importantly new games as they are produced. This new data and analysis could then be contrasted against this work to see how it compares to other games that were not previously used.

A group could also go as far as attempting to produce a group server browser for a single game or a range of games that uses the best server metric that we developed. This server browser could then be tested with various sized groups to test its effectiveness and ask the participants to gauge how effective they perceived the server browser was. The most difficult aspect of this project would most likely be finding a willing group of

participants to test this on and more importantly participants that are far enough from each other for it to be an effective test.

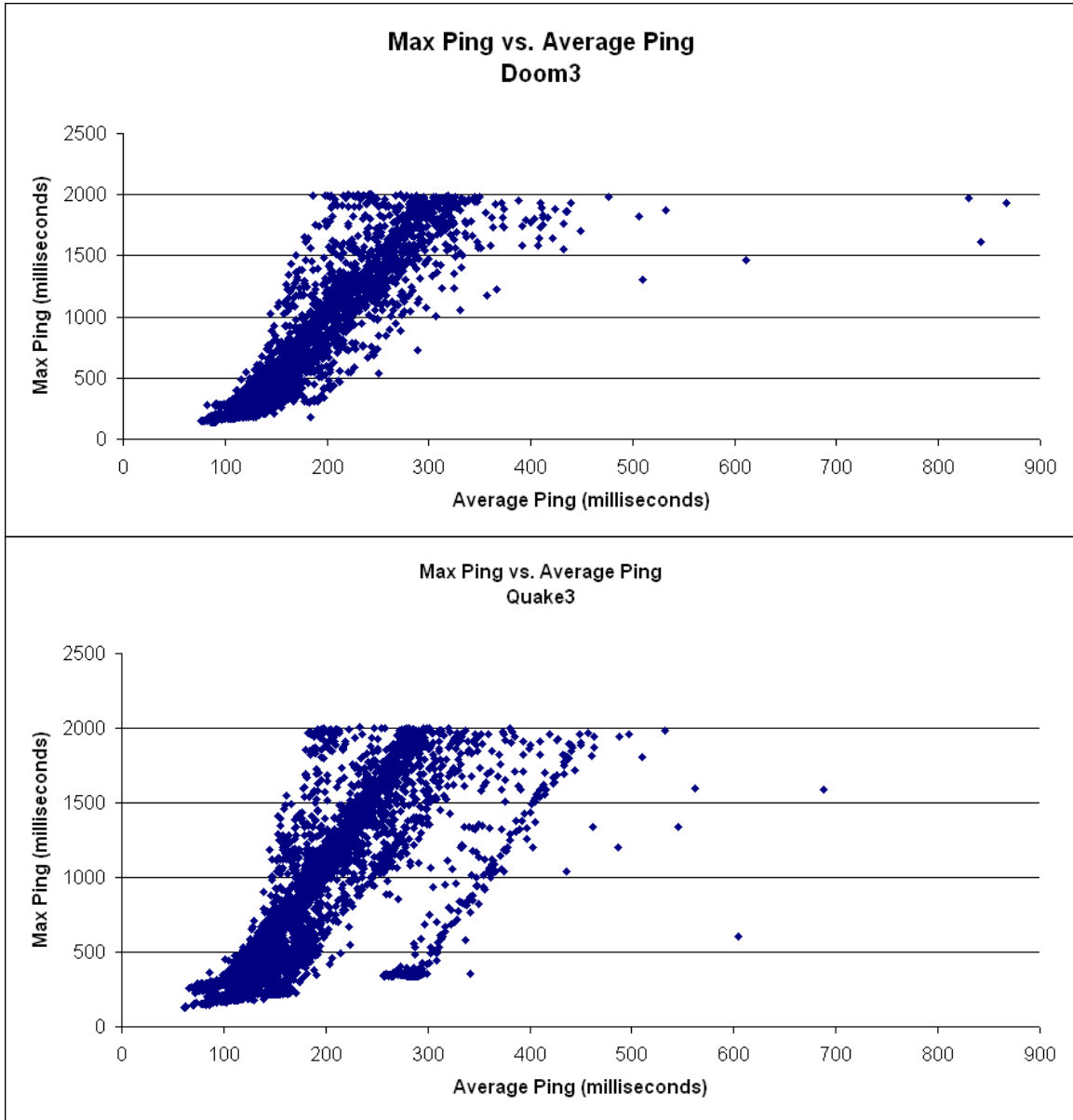
Finally, a single player server browser could be developed for the sake of gathering data on players' habits when it comes to the servers they are shown and the ones they actually play on. Data could be gathered on how many servers a player connects to before settling on one to actually play on. This data, with the pings of those servers when they were queried before being picked, would be useful in analyzing a player's method of choosing a server. This could then be used to devise a far more productive method of picking a server than one that was calculated from just query data.

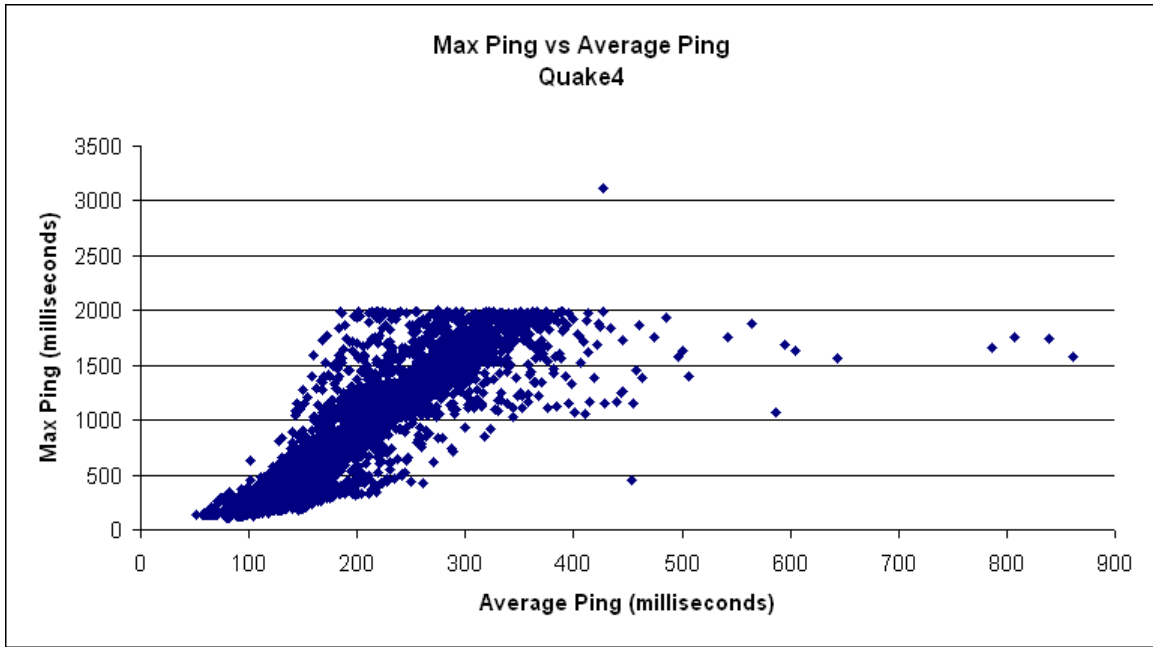
7 References

- [1] Gargolinski, S., St. Pierre, C. & Claypool, M. (2005, Oct). [Game server selection for multiple players](#). *Proceedings of the 4th ACM Network and System Support for Games (NetGames)*, Hawthorne, N.Y.
- [2] Branch, P. A. & Armitage, G. J. (2005, Dec). Towards a general model of first person shooter game traffic. *In Proc. Workshop on Network and System Support for Games*.
- [3] Lang, T. & Armitage, G. J. (2003, June). A NS2 model for the Xbox system link game Halo. *In Proc. Workshop on Network and System Support for Games*.
- [4] Dick, M., Wellnitz, O. & Wolf, L. (2005). Analysis of factors affecting players' performance and perception in multiplayer games. *In Proc. Network and System Support for Games*.
- [5] Chambers, C., Feng, W. & Feng, W. (2003). A geographic redirection service for on-line games. *Proc. ACM International Conference on Multimedia*.
- [6] [QStat-Real-Time Game Server Stats](#). Open Source Technology Group. Retrieved Apr 20, 2007 from Website: <http://sourceforge.net/projects/qstat>
- [7] [Planet Lab: An Open Platform for Developing, Deploying and Accessing Planetary-Scale Services](#). The Trustees of Princeton University. Retrieved Apr 18, 2007, from Website: <http://www.planet-lab.org/>
- [8] [CoMon – A monitoring Infrastructure for Planet Lab](#). The Trustees of Princeton University. Retrieved Apr 22, 2007, from Website: <http://comon.cs.princeton.edu/>
- [9] Brun, J., Safaei, F. & Boustead, P. (2006). Server topology in online games. *Proc. Workshop on Network and System Support for Games*.
- [10] Brun, J., Safaei, F. & Boustead, P. (2006, Jan). Fairness and playability in online multiplayer games. *IEEE CCNC/NIME*.

Appendix A

A1 Max ping vs. Average Ping Graphs





A2 Master Server List

Doom3

idnet.ua-corp.com

Quake3

master3.idsoftware.com

Quake4

q4master.idsoftware.com

Return to Castle Wolfenstein

Wolfmaster.idsoftware.com

A3 Game Server List

Doom3

88.149.151.205:27666

81.169.141.133:27666

76.171.209.98:27666

74.96.82.43:27666

72.9.107.234:27666

72.36.157.157:27666

72.36.133.230:27960

70.86.179.3:27666

69.65.15.138:27666

68.42.196.9:27666

68.232.163.54:27666

64.92.172.58:27666

64.71.180.29:27669

64.34.240.135:28004
64.34.200.229:27667
64.34.200.229:27666
64.182.169.193:27960
24.185.116.227:27666
216.180.251.178:27666
195.34.7.33:27666

Quake3

213.190.3.4:27960
67.43.163.194:27960
89.174.246.19:27990
85.131.137.51:27015
83.133.127.23:27973
82.99.173.139:27962
82.99.173.139:27961
82.99.173.139:27960
81.27.73.100:27965
72.51.60.107:27960
70.86.44.154:27975
70.84.181.109:27960
70.84.181.106:27960
69.93.136.70:27960
69.90.34.22:27960
67.43.163.194:27960
64.255.174.111:27960
212.54.98.123:27960
210.55.92.68:27960
12.6.136.34:27960

Quake4

88.149.151.201:22003
8.2.122.174:28004
64.182.153.70:28004
213.208.119.120:28004
70.105.163.126:28004
69.12.9.94:28004
213.208.119.120:28007
213.208.119.120:28006
195.238.1.26:27672
193.71.196.49:28007
140.192.244.114:28005
85.21.37.19:28005
8.9.9.108:28004
8.3.5.13:28004
69.65.32.192:28004
66.55.159.133:28004
195.13.62.5:28700
195.13.62.44:27700
84.244.186.42:28004
209.40.98.113:28004

Return to Castle Wolfenstein

89.106.68.16:27961
85.131.136.97:27015
84.107.145.237:27960
82.192.78.33:27960

82.192.78.2:27960
69.65.31.90:27960
69.65.31.65:27960
67.18.221.194:27970
62.75.174.216:27960
62.173.168.2:27010
208.179.134.251:27960
85.31.186.218:27970
74.52.140.149:27960
70.86.152.20:27961
69.65.31.67:27960
67.18.216.130:27960
195.4.19.211:27960
84.244.156.129:27960
82.192.78.61:27970
67.18.221.196:27970

A4 Client List by Country

USA

75-130-96-12.static.oxfr.ma.charter.com

75-130-96-13.static.oxfr.ma.charter.com

planetlab6.millennium.berkeley.edu

kc-sce-plab1.umkc.edu

kc-sce-plab2.umkc.edu

planetlab1.csee.usf.edu

planetlab2.csee.usf.edu

Puerto Rico

node1.planetlab.uprr.pr

node2.planetlab.uprr.pr

Israel

planet1.cs.huji.ac.il

planet5.cs.huji.ac.il

United Kingdom

planet1.manchester.ac.uk

planetlab1.xeno.cl.cam.ac.uk

Japan

planetlab-02.kusa.ac.jp

Germany

planetlab01.mpi-sws.mpg.de

planetlab02.mpi-sws.mpg.de

planetlab04.mpi-sws.mpg.de

France

planetlab1.inria.fr

planetlab3.inria.fr

Spain

planetlab1.ls.fi.upm.es

planetlab2.ls.fi.upm.es

Korea

planetlab1.netmedia.gist.ac.kr

planetlab2.netmedia.gist.ac.kr

Australia

plnode01.cs.mu.oz.au

plnode02.cs.mu.oz.au



Figure 7-1: World Map with Nodes represented by blue dots.

A5 Software

A5.1 Organizer Script

```
#!/usr/bin/perl
#program that uses Expect to automate process of gathering qstat data from multiple
clients

use lib "/home/$ENV{'USER'}/mqp/perl/lib/perl5/",
"/home/$ENV{'USER'}/mqp/perl/lib/perl5/site_perl/";
use Expect;
use strict;
#$Expect::Debug = 2;
$|++;

if(scalar @ARGV < 3)
{
    print "Usage: $0 <client info file> <game type> <server1, server2, ...>\n";
    exit;
}

my $deniedfile = "denied.txt";
my $timeoutfile = "timeout.txt";
my $noconnectfile = "noconnect.txt";
my $timeout = 20;
my ($clientinfo, $game, $servers)= @ARGV;
```

```

my %clientexps, my %clientprompts;
my(undef, $min, $hour, $day, $month, undef, undef, undef) = localtime(time);
$min = sprintf("%02d", $min);
$hour = sprintf("%02d", $hour);
$day = sprintf("%02d", $day);
$month = sprintf("%02d", $month + 1);

open CLIENT, "<$clientinfo" or die "Can't open $clientinfo: $!\n";
my @clients = <CLIENT>;
close CLIENT;

open(DENIED, ">$deniedfile") or die "Can't open $deniedfile $!\n";
open(TIMEDOUT, ">$timedoutfile") or die "Can't open $timedoutfile $!\n";
open(NOCONNECT, ">$noconnectfile") or die "Can't open $noconnectfile $!\n";

LOOP: foreach my $client (@clients)
{
    chomp($client);
    my $username = '<PLusername>';
    my $password = '<PLpassword>';
    my $prompt = "$username@.((split(/\.\/, $client))[0]);
    print "|$client| |$username| |$password| |$prompt|\n";
    print "\nConnecting to $client...\n";
    my $tempexp = Expect->spawn("ssh -i ~/.ssh/id_rsa ".$username."@".$client)
        or die "Couldn't connect to client $client\n";
    $tempexp->expect(undef, [qr/passphrase.*:/i,
        sub {
            print $tempexp "$password\r";
            print "\nentered password\n";
            exp_continue;
        }],
        [qr/\(yes\/no\/)\?/i,
        sub {
            print $tempexp "yes\r";
            print "\nsaid yes\n";
            exp_continue;
        }],
        [qr/Connection timed out/i,
        sub {
            print "Client timed out, removing from list\n";
            print TIMEDOUT "$client\n";
            next LOOP;
        }],
        [qr/REMOTE HOST IDENTIFICATION HAS CHANGED/i,
        sub {
            print "\nRSA key for client changed, removing from list of known
hosts and readding\n";
            my $tempprompt = 'bash';
            my $exp = Expect->spawn("bash");
            $exp->expect(10, $tempprompt);
            print $exp "grep -v $client ~/.ssh/known_hosts > ~/.ssh/temp\r";
            $exp->expect(10, $tempprompt);
            print $exp "mv ~/.ssh/temp ~/.ssh/known_hosts\r";
            $exp->expect(10, $tempprompt);
            redo LOOP;
        }],
        [qr/Permission denied/i,
        sub {
            print "Client doesn't have correct credentials,\nremoving from
list\n";
            print DENIED "$client\n";
            next LOOP;
        }],
        [qr/Temporary failure in name resolution/i,
        sub{
            print "Can't reach client for some reason\n";
            print NOCONNECT "$client\n";
            next LOOP;
        }],
        [qr/Connection closed/i,
        sub{

```

```

        print "Can't reach client for some reason\n";
        print NOCONNECT "$client\n";
        next LOOP;
    }],
    [qr/Connection refused/i,
    sub{
        print "Can't reach client for some reason\n";
        print NOCONNECT "$client\n";
        next LOOP;
    }],
    '-re', qr/$prompt/)
    or warn "Did not get prompt for password or command prompt\n";
    $clientexps{$client} = $tempexp;
    $clientprompts{$client} = $prompt;
}
print "\nDEBUG: **|Clients logged in|**\n";

foreach my $key (sort keys %clientexps)
{
    my $exp = $clientexps{$key};
    my $prompt = $clientprompts{$key};
    print $exp "./gatherer.pl $prompt $game $month$day $hour$min $servers,127.0.0.1\r";
}
print "\nDEBUG: **|Client scripts started|**\n";

my $attempts = 0;
foreach my $key (sort keys %clientexps)
{
    my $exp = $clientexps{$key};
    my $prompt = $clientprompts{$key};
    while(!$exp->expect(2, "DONE QUERYING SERVERS!!") . $attempts < 180)
    {
        $attempts++;
        sleep(3);
    }
    if($attempts >= 180)
    {
        print STDERR "Client $key is having problems, killing process";
        print $exp "\x03";
        $exp->expect($timeout, '-re', $prompt);
        print $exp "exit\r";
        delete $clientexps{$key};
    }
    else
    {
        print "\nDEBUG: **|" . ($exp->exp_after()) . "|**\n";
    }
}
print "\nDEBUG: **|Client scripts finished|**\n";

LOOPB: foreach my $key (sort keys %clientexps)
{
    my $exp = $clientexps{$key};
    my $prompt = $clientprompts{$key};
    print "\nDEBUG: **|Getting data from Client $key|**\n";
    print $exp "scp ~/qstat-\t/". $game . "xml <username>@\<host>:~/mqp/data/\r";
    $exp->expect(undef, [qr/password:/i,
        sub {
            print $exp "<password>\r";
            print "\nentered password\n";
            exp_continue;
        }],
    [qr/\(yes\/no\/)\?/i,
    sub {
        print $exp "yes\r";
        print "\nsaid yes\n";
        exp_continue;
    }],
    [qr/Connection timed out/i,
    sub {
        print "Client timed out, removing from list\n";

```



```

        print TIMEDOUT "<host>\n";
        next LOOPB;
    }],
    [qr/REMOTE HOST IDENTIFICATION HAS CHANGED/i,
    sub {
        print "\nRSA key for client changed, removing from list of known hosts
and reading\n";
        my $tempprompt = 'bash';
        my $tempexp = Expect->spawn("bash");
        $exp->expect(10, $tempprompt);
        print $tempexp "grep -v <host> ~/.ssh/known_hosts > ~/.ssh/temp\r";
        $exp->expect(10, $tempprompt);
        print $tempexp "mv ~/.ssh/temp ~/.ssh/known_hosts\r";
        $exp->expect(10, $tempprompt);
        redo LOOPB;
    }],
    [qr/Permission denied/i,
    sub {
        print "Client doesn't have correct credentials,\nremoving from list\n";
        print DENIED "<host>\n";
        next LOOPB;
    }],
    [qr/Temporary failure in name resolution/i,
    sub{
        print "Can't reach client for some reason\n";
        print NOCONNECT "<host>\n";
        next LOOPB;
    }],
    [qr/Connection closed/i,
    sub{
        print "Can't reach client for some reason\n";
        print NOCONNECT "<host>\n";
        next LOOPB;
    }],
    '-re', qr/$prompt/);
print $exp "rm -rf $game*.xml\r";
$exp->expect($timeout, '-re', $prompt);
}

```

A5.2 Gatherer Script

```

#!/usr/bin/perl
#program that uses Expect to automate process of using qstat to gather data

use lib "/home/$ENV{'USER'}/mqp/perl/lib/perl5/",
"/home/$ENV{'USER'}/mqp/perl/lib/perl5/site_perl/";
use Expect;
use strict;
$|++;

$SIG{'INT'} = 'CLEANUP';

if(scalar @ARGV < 4)
{
    print "Usage: $0 <prompt RE> <game type> <date (mmd)> <time (HH)> <server1, server2,
...>\n";
    exit;
}

my $timeout = 30;
my($prompt, $game, $date, $time, $list) = @ARGV;
my @address = split(/,/, $list);

my $exp = Expect->spawn("bash") or die "Unable to open new command line: $!\n";
$exp->expect(10, '-re', $prompt);

print $exp "cd ~/qstat-2.11\r";
$exp->expect(10, 'cd ~/qstat-2.11');
$exp->expect(10, '-re', $prompt);

```

```

foreach my $address (@address)
{
    print $exp "./qstat -cfg ../qstat.cfg -P -R -xml -progress -of
    ".$game."_".$address."_".$date."_".$time."_".$ENV{'HOSTNAME'}.".xml - $game $address\r";
    sleep 3;
    my @result = $exp->expect(undef, '-re', $prompt);
    print "\n\nDEBUG: ***|@result|***\n\n";
}
print $exp "ps -adef | grep qstat\r";
$exp->expect(undef, $prompt);

print "DONE QUERYING SERVERS!!";

sub CLEANUP
{
    print $exp "\x03";
    $exp->expect(undef, '-re', $prompt);
    print $exp "rm -vf $game"."*_.$date"."_.$time"."_".$ENV{'HOSTNAME'}.".xml \r";
    $exp->expect(5, '-re', $prompt);
    exit;
}

```

A6 Sample XML data files

A6.1 Master Server Data

Query Sample #1

```

<?xml version="1.0" encoding="iso-8859-1"?>
<qstat>
  <server type="DM3M" address="idnet.ua-corp.com" status="UP" servers="75">
  </server>
  <server type="DM3S" address="72.9.107.234:27666" status="UP">
    <hostname>72.9.107.234:27666</hostname>
    <name> cd\...c: Clan</name>
    <gametype></gametype>
    <map>game/mp/d3dm4</map>
    <numplayers>0</numplayers>
    <maxplayers>8</maxplayers>
    <ping>10</ping>
    <retries>0</retries>
  </server>
  <server type="DM3S" address="64.92.172.58:27666" status="UP">
    <hostname>64.92.172.58:27666</hostname>
    <name>HBB-clan.com</name>
    <gametype>lms3</gametype>
    <map>game/hell1</map>
    <numplayers>0</numplayers>
    <maxplayers>8</maxplayers>
    <ping>49</ping>
    <retries>0</retries>
  </server>
  <server type="DM3S" address="72.36.157.157:27666" status="UP">
    <hostname>72.36.157.157:27666</hostname>
    <name>SSDoomClan.com EMZ</name>
    <gametype>EMZ</gametype>
    <map>swd3dml</map>
    <numplayers>0</numplayers>
    <maxplayers>6</maxplayers>
    <ping>52</ping>
    <retries>0</retries>
  </server>
  <server type="DM3S" address="72.226.230.127:27666" status="UP">
    <hostname>72.226.230.127:27666</hostname>
    <name>Hobbes LMS 3.0 Co-op Server</name>
    <gametype>lms3</gametype>
    <map>game/mars_city1</map>
    <numplayers>0</numplayers>
    <maxplayers>4</maxplayers>
    <ping>32</ping>

```

```

        <retries>0</retries>
</server>
<server type="DM3S" address="212.227.60.177:27666" status="UP">
  <hostname>212.227.60.177:27666</hostname>
  <name>OpenCoop 1.1</name>
  <gametype>opencoop</gametype>
  <map>game/hellhole</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>95</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="68.232.163.54:27666" status="UP">
  <hostname>68.232.163.54:27666</hostname>
  <name>-\\BAD Company//-</name>
  <gametype>emz</gametype>
  <map>swd3dm2</map>
  <numplayers>0</numplayers>
  <maxplayers>8</maxplayers>
  <ping>44</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="216.21.45.36:27666" status="UP">
  <hostname>216.21.45.36:27666</hostname>
  <name>Hells Kitchen ChicagoROE</name>
  <gametype>lms3</gametype>
  <map>game/deltax</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>34</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.182.158.145:27970" status="UP">
  <hostname>64.182.158.145:27970</hostname>
  <name>{{The Sandwich Shop}}</name>
  <gametype></gametype>
  <map>aerod3</map>
  <numplayers>0</numplayers>
  <maxplayers>6</maxplayers>
  <ping>26</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="216.180.251.178:27666" status="UP">
  <hostname>216.180.251.178:27666</hostname>
  <name>www.eXg-clan.com [PRI]</name>
  <gametype>emz</gametype>
  <map>game/mp/d3refrag</map>
  <numplayers>0</numplayers>
  <maxplayers>8</maxplayers>
  <ping>34</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="68.42.196.9:27666" status="UP">
  <hostname>68.42.196.9:27666</hostname>
  <name>CoZmicShReddeR.com Clan_C7 NO_CLAN_TAGS</name>
  <gametype>lms3</gametype>
  <map>game/erebus6</map>
  <numplayers>0</numplayers>
  <maxplayers>6</maxplayers>
  <ping>41</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="69.13.26.28:27666" status="UP">
  <hostname>69.13.26.28:27666</hostname>
  <name>Nonameservers</name>
  <gametype></gametype>
  <map>game/mp/d3dm2</map>
  <numplayers>0</numplayers>
  <maxplayers>16</maxplayers>
  <ping>26</ping>
  <retries>0</retries>

```

```

</server>
<server type="DM3S" address="81.27.73.100:27800" status="UP">
  <hostname>81.27.73.100:27800</hostname>
  <name>the-emz.com Official EMZ v.1.30 (Beta 3) - lazeryattack.com</name>
  <gametype>emz</gametype>
  <map>aerod3</map>
  <numplayers>1</numplayers>
  <maxplayers>4</maxplayers>
  <ping>80</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.34.200.229:27668" status="UP">
  <hostname>64.34.200.229:27668</hostname>
  <name>FragArcade.com - Live Stats - #692</name>
  <gametype></gametype>
  <map>game/mp/d3dm5</map>
  <numplayers>0</numplayers>
  <maxplayers>10</maxplayers>
  <ping>21</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.34.200.229:27667" status="UP">
  <hostname>64.34.200.229:27667</hostname>
  <name>FragArcade.com - Fragit - #800</name>
  <gametype></gametype>
  <map>game/mp/d3dm2</map>
  <numplayers>0</numplayers>
  <maxplayers>10</maxplayers>
  <ping>21</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="88.149.151.205:27666" status="UP">
  <hostname>88.149.151.205:27666</hostname>
  <name>NGI [EMZ 1.21] Doom 3 Generic #1</name>
  <gametype>emz</gametype>
  <map>game/mp/d3dm3</map>
  <numplayers>3</numplayers>
  <maxplayers>4</maxplayers>
  <ping>107</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="62.4.81.246:30100" status="UP">
  <hostname>62.4.81.246:30100</hostname>
  <name>Doom3Maps.org - EMZ 1.21 - Server 1 by Stormix</name>
  <gametype>emz</gametype>
  <map>game/mp/d3dm2</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>102</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.71.180.29:27671" status="UP">
  <hostname>64.71.180.29:27671</hostname>
  <name>Last Stronghold (CTF)</name>
  <gametype>d3xp</gametype>
  <map>game/mp/d3ctf1</map>
  <numplayers>0</numplayers>
  <maxplayers>8</maxplayers>
  <ping>85</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="195.13.62.54:28100" status="UP">
  <hostname>195.13.62.54:28100</hostname>
  <name>Team-Nonpareil.com|EMZ v.1.30 (beta3)</name>
  <gametype>emz</gametype>
  <map>game/mp/d3dm2</map>
  <numplayers>0</numplayers>
  <maxplayers>8</maxplayers>
  <ping>98</ping>
  <retries>0</retries>
</server>

```

```

<server type="DM3S" address="74.96.82.43:27666" status="UP">
  <hostname>74.96.82.43:27666</hostname>
  <name>Warlord's Arena II</name>
  <gametype></gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>41</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="71.139.67.248:27666" status="UP">
  <hostname>71.139.67.248:27666</hostname>
  <name>2007! Halo 3 YEAR :D</name>
  <gametype>opencoop</gametype>
  <map>game/site3</map>
  <numplayers>2</numplayers>
  <maxplayers>3</maxplayers>
  <ping>123</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="69.65.15.138:27666" status="UP">
  <hostname>69.65.15.138:27666</hostname>
  <name>[GUI] CLAN PURE SERVER</name>
  <gametype>d3xp</gametype>
  <map>game/mp/d3xpdml</map>
  <numplayers>0</numplayers>
  <maxplayers>8</maxplayers>
  <ping>23</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="72.36.133.230:27960" status="UP">
  <hostname>72.36.133.230:27960</hostname>
  <name>Smokes house v 2.1</name>
  <gametype></gametype>
  <map>dc/mp/dc_snipertest</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>49</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="81.169.141.133:27666" status="UP">
  <hostname>81.169.141.133:27666</hostname>
  <name>Doom 3</name>
  <gametype></gametype>
  <map>game/mp/d3dml</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>125</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.182.169.193:27960" status="UP">
  <hostname>64.182.169.193:27960</hostname>
  <name>pineapple happynewyear!</name>
  <gametype></gametype>
  <map>dead_dust</map>
  <numplayers>6</numplayers>
  <maxplayers>8</maxplayers>
  <ping>9</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="213.190.46.52:62153" status="DOWN">
  <hostname>213.190.46.52:62153</hostname>
</server>
<server type="DM3S" address="24.129.21.245:62534" status="UP">
  <hostname>24.129.21.245:62534</hostname>
  <name>Meet Hell</name>
  <gametype></gametype>
  <map>game/mp/d3dml</map>
  <numplayers>0</numplayers>
  <maxplayers>6</maxplayers>
  <ping>55</ping>

```

```

        <retries>0</retries>
</server>
<server type="DM3S" address="62.4.81.252:30700" status="UP">
  <hostname>62.4.81.252:30700</hostname>
  <name>Stormix.de Doom 3 Testserver</name>
  <gametype></gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>101</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="81.169.167.62:27666" status="UP">
  <hostname>81.169.167.62:27666</hostname>
  <name>Dim-ension.de-OpenCoop#1</name>
  <gametype>opencoop</gametype>
  <map>game/alphalabs4</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>118</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="69.225.52.201:27666" status="UP">
  <hostname>69.225.52.201:27666</hostname>
  <name>Dept.ofCorrections</name>
  <gametype></gametype>
  <map>game/mp/d3dm5</map>
  <numplayers>0</numplayers>
  <maxplayers>8</maxplayers>
  <ping>102</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.34.200.229:27666" status="UP">
  <hostname>64.34.200.229:27666</hostname>
  <name>FragArcade.com - Live Stats - #692</name>
  <gametype></gametype>
  <map>game/mp/d3dm2</map>
  <numplayers>0</numplayers>
  <maxplayers>10</maxplayers>
  <ping>22</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.71.180.29:27667" status="UP">
  <hostname>64.71.180.29:27667</hostname>
  <name>Alienware.com - Server III</name>
  <gametype></gametype>
  <map>game/mp/d3dm1</map>
  <numplayers>0</numplayers>
  <maxplayers>8</maxplayers>
  <ping>85</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="68.196.190.90:27666" status="UP">
  <hostname>68.196.190.90:27666</hostname>
  <name>oes Coop Server</name>
  <gametype>lms3</gametype>
  <map>game/alphalabs2</map>
  <numplayers>2</numplayers>
  <maxplayers>2</maxplayers>
  <ping>21</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="194.65.14.105:27666" status="UP">
  <hostname>194.65.14.105:27666</hostname>
  <name>[ SAPOPlay | WAR-ZONE ] Tourney #1</name>
  <gametype></gametype>
  <map>game/mp/d3dm2</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>138</ping>
  <retries>0</retries>

```

```

</server>
<server type="DM3S" address="85.214.55.250:27666" status="UP">
  <hostname>85.214.55.250:27666</hostname>
  <name>Dim-ension.de-OpenCoop#2</name>
  <gametype>opencoop</gametype>
  <map>game/hellhole</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>132</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="195.34.7.33:27666" status="UP">
  <hostname>195.34.7.33:27666</hostname>
  <name>StreamArena Doom3 server</name>
  <gametype></gametype>
  <map>game/mp/d3dm3</map>
  <numplayers>0</numplayers>
  <maxplayers>12</maxplayers>
  <ping>136</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="76.171.209.98:27666" status="UP">
  <hostname>76.171.209.98:27666</hostname>
  <name>QnDfix LMScoop! </name>
  <gametype>lms3</gametype>
  <map>game/alphalabs1</map>
  <numplayers>0</numplayers>
  <maxplayers>6</maxplayers>
  <ping>99</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="130.88.189.64:27666" status="UP">
  <hostname>130.88.189.64:27666</hostname>
  <name>Man Uni: Doom 3</name>
  <gametype>base</gametype>
  <map>game/mp/d3dm1</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>108</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="82.18.109.174:27666" status="DOWN">
  <hostname>82.18.109.174:27666</hostname>
</server>
<server type="DM3S" address="70.86.179.2:27666" status="UP">
  <hostname>70.86.179.2:27666</hostname>
  <name>Alienware.com - Server I</name>
  <gametype></gametype>
  <map>game/mp/d3dm3</map>
  <numplayers>0</numplayers>
  <maxplayers>8</maxplayers>
  <ping>52</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="194.65.14.105:27999" status="UP">
  <hostname>194.65.14.105:27999</hostname>
  <name>[ SAPOPlay | WAR-ZONE ] Tourney #2</name>
  <gametype></gametype>
  <map>game/mp/d3dm2</map>
  <numplayers>1</numplayers>
  <maxplayers>4</maxplayers>
  <ping>137</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="83.250.130.47:27666" status="UP">
  <hostname>83.250.130.47:27666</hostname>
  <name>NEGERKUKENS .TERKOMST!!!</name>
  <gametype>opencoop</gametype>
  <map>game/delta2a</map>
  <numplayers>2</numplayers>
  <maxplayers>4</maxplayers>

```

```

        <ping>120</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="62.4.81.249:30000" status="UP">
        <hostname>62.4.81.249:30000</hostname>
        <name>Doom3Maps.org - EMZ 1.21 - Server 3 by Stormix</name>
        <gametype>emz</gametype>
        <map>swd3dm1</map>
        <numplayers>0</numplayers>
        <maxplayers>4</maxplayers>
        <ping>102</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="74.96.82.43:27668" status="UP">
        <hostname>74.96.82.43:27668</hostname>
        <name>Warlord's Arena</name>
        <gametype></gametype>
        <map>game/mp/d3dm5</map>
        <numplayers>4</numplayers>
        <maxplayers>4</maxplayers>
        <ping>42</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="64.182.158.105:27970" status="UP">
        <hostname>64.182.158.105:27970</hostname>
        <name>www.eXg-clan.com [CHI]</name>
        <gametype>emz</gametype>
        <map>swd3dm2</map>
        <numplayers>0</numplayers>
        <maxplayers>8</maxplayers>
        <ping>27</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="64.71.180.29:27666" status="UP">
        <hostname>64.71.180.29:27666</hostname>
        <name>Slaughter House Dm...</name>
        <gametype></gametype>
        <map>aerod3</map>
        <numplayers>0</numplayers>
        <maxplayers>8</maxplayers>
        <ping>86</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="70.86.179.3:27666" status="UP">
        <hostname>70.86.179.3:27666</hostname>
        <name>Alienware.com - Server II</name>
        <gametype></gametype>
        <map>game/mp/d3dm3</map>
        <numplayers>2</numplayers>
        <maxplayers>8</maxplayers>
        <ping>47</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="200.171.26.133:39883" status="UP">
        <hostname>200.171.26.133:39883</hostname>
        <name>!!CYBERMAC BRAZIL!!</name>
        <gametype></gametype>
        <map>game/mp/d3dm3</map>
        <numplayers>0</numplayers>
        <maxplayers>4</maxplayers>
        <ping>172</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="84.193.214.108:60054" status="UP">
        <hostname>84.193.214.108:60054</hostname>
        <name>Metalex</name>
        <gametype></gametype>
        <map>game/mp/d3dm2</map>
        <numplayers>2</numplayers>
        <maxplayers>4</maxplayers>
        <ping>115</ping>

```



```

        <retries>0</retries>
</server>
<server type="DM3S" address="85.140.120.11:27666" status="UP">
  <hostname>85.140.120.11:27666</hostname>
  <name> </name>
  <gametype></gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>1</numplayers>
  <maxplayers>7</maxplayers>
  <ping>158</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="65.98.61.18:27666" status="UP">
  <hostname>65.98.61.18:27666</hostname>
  <name>Generation {XS} EMZ Server</name>
  <gametype>emz</gametype>
  <map>game/mp/d3dm5</map>
  <numplayers>0</numplayers>
  <maxplayers>6</maxplayers>
  <ping>8</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.71.180.29:27670" status="UP">
  <hostname>64.71.180.29:27670</hostname>
  <name>Slaughter House Tourney..</name>
  <gametype></gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>0</numplayers>
  <maxplayers>8</maxplayers>
  <ping>85</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="195.4.104.156:27666" status="UP">
  <hostname>195.4.104.156:27666</hostname>
  <name>gzdoom</name>
  <gametype></gametype>
  <map>game/mp/d3dm1</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>111</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="86.122.226.161:27666" status="UP">
  <hostname>86.122.226.161:27666</hostname>
  <name>DOOM SERVER</name>
  <gametype></gametype>
  <map>game/mp/d3dm3</map>
  <numplayers>2</numplayers>
  <maxplayers>2</maxplayers>
  <ping>151</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="85.31.186.170:28000" status="UP">
  <hostname>85.31.186.170:28000</hostname>
  <name>RoseBuds Doom3</name>
  <gametype></gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>0</numplayers>
  <maxplayers>12</maxplayers>
  <ping>110</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="128.174.157.33:27666" status="UP">
  <hostname>128.174.157.33:27666</hostname>
  <name>BenZone Doom3 CoOp</name>
  <gametype>lms3</gametype>
  <map>game/mc_underground</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>38</ping>
  <retries>0</retries>

```

```

</server>
<server type="DM3S" address="216.21.45.37:27666" status="UP">
  <hostname>216.21.45.37:27666</hostname>
  <name>Hells Kitchen Chicago</name>
  <gametype>lms3</gametype>
  <map>game/alphalabs3</map>
  <numplayers>0</numplayers>
  <maxplayers>6</maxplayers>
  <ping>34</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="203.135.193.176:27666" status="DOWN">
  <hostname>203.135.193.176:27666</hostname>
</server>
<server type="DM3S" address="82.47.44.243:27666" status="UP">
  <hostname>82.47.44.243:27666</hostname>
  <name>ultimate fight II</name>
  <gametype></gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>3</numplayers>
  <maxplayers>4</maxplayers>
  <ping>103</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="216.52.148.125:27666" status="UP">
  <hostname>216.52.148.125:27666</hostname>
  <name>s3lf Pvt Tourney (chic)</name>
  <gametype></gametype>
  <map>swd3dm2</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>23</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="62.4.74.149:30100" status="UP">
  <hostname>62.4.74.149:30100</hostname>
  <name>||PAIN</name>
  <gametype></gametype>
  <map>game/mp/d3dm1</map>
  <numplayers>4</numplayers>
  <maxplayers>4</maxplayers>
  <ping>97</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="85.70.106.214:27666" status="UP">
  <hostname>85.70.106.214:27666</hostname>
  <name>ITCH</name>
  <gametype>d3xp</gametype>
  <map>game/mp/d3xpdm4</map>
  <numplayers>1</numplayers>
  <maxplayers>4</maxplayers>
  <ping>168</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.34.240.135:28004" status="UP">
  <hostname>64.34.240.135:28004</hostname>
  <name>The Gauntlet. Q4Max0.77a Chicago</name>
  <gametype>q4max</gametype>
  <map>mp/swq4dm1</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>24</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="62.85.122.55:27666" status="UP">
  <hostname>62.85.122.55:27666</hostname>
  <name>DOOM Server</name>
  <gametype></gametype>
  <map>game/mp/d3dm2</map>
  <numplayers>1</numplayers>
  <maxplayers>4</maxplayers>

```

```

        <ping>123</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="85.14.217.81:30000" status="UP">
        <hostname>85.14.217.81:30000</hostname>
        <name>TsaiTschen Q4</name>
        <gametype>q4base</gametype>
        <map>mp/q4dm5</map>
        <numplayers>0</numplayers>
        <maxplayers>12</maxplayers>
        <ping>98</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="64.71.180.29:27669" status="UP">
        <hostname>64.71.180.29:27669</hostname>
        <name>Slaughter House Team Dm...</name>
        <gametype></gametype>
        <map>swd3dml</map>
        <numplayers>0</numplayers>
        <maxplayers>8</maxplayers>
        <ping>86</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="82.147.186.224:27666" status="UP">
        <hostname>82.147.186.224:27666</hostname>
        <name>Canta server !</name>
        <gametype></gametype>
        <map>game/mp/d3dm2</map>
        <numplayers>5</numplayers>
        <maxplayers>8</maxplayers>
        <ping>191</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="85.99.200.8:20758" status="TIMEOUT">
        <hostname>85.99.200.8:20758</hostname>
    </server>
    <server type="DM3S" address="217.147.80.89:27667" status="TIMEOUT">
        <hostname>217.147.80.89:27667</hostname>
    </server>
    <server type="DM3S" address="219.241.229.163:27666" status="TIMEOUT">
        <hostname>219.241.229.163:27666</hostname>
    </server>
    <server type="DM3S" address="87.179.102.65:32876" status="TIMEOUT">
        <hostname>87.179.102.65:32876</hostname>
    </server>
    <server type="DM3S" address="121.247.113.147:61253" status="TIMEOUT">
        <hostname>121.247.113.147:61253</hostname>
    </server>
    <server type="DM3S" address="172.207.56.101:27666" status="TIMEOUT">
        <hostname>172.207.56.101:27666</hostname>
    </server>
    <server type="DM3S" address="85.96.188.74:27666" status="TIMEOUT">
        <hostname>85.96.188.74:27666</hostname>
    </server>
    <server type="DM3S" address="85.140.166.202:27666" status="TIMEOUT">
        <hostname>85.140.166.202:27666</hostname>
    </server>
    <server type="DM3S" address="84.247.57.39:27666" status="TIMEOUT">
        <hostname>84.247.57.39:27666</hostname>
    </server>
</qstat>

```

Query sample #2

```

<?xml version="1.0" encoding="iso-8859-1"?>
<qstat>
    <server type="DM3M" address="idnet.ua-corp.com" status="UP" servers="95">
    </server>
    <server type="DM3S" address="72.9.107.234:27666" status="UP">
        <hostname>72.9.107.234:27666</hostname>
        <name> cd\...c: Clan</name>
        <gametype></gametype>
    </server>

```

```

        <map>swd3dm1</map>
        <numplayers>1</numplayers>
        <maxplayers>8</maxplayers>
        <ping>10</ping>
        <retries>0</retries>
</server>
<server type="DM3S" address="72.36.157.157:27666" status="UP">
  <hostname>72.36.157.157:27666</hostname>
  <name>SSDoomClan.com EMZ</name>
  <gametype>EMZ</gametype>
  <map>game/mp/epic</map>
  <numplayers>1</numplayers>
  <maxplayers>6</maxplayers>
  <ping>52</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="212.227.60.177:27666" status="UP">
  <hostname>212.227.60.177:27666</hostname>
  <name>OpenCoop 1.1</name>
  <gametype>opencoop</gametype>
  <map>game/hellhole</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>95</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="216.21.45.36:27666" status="UP">
  <hostname>216.21.45.36:27666</hostname>
  <name>Hells Kitchen ChicagoROE</name>
  <gametype>lms3</gametype>
  <map>game/hell</map>
  <numplayers>3</numplayers>
  <maxplayers>4</maxplayers>
  <ping>35</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="216.180.251.178:27666" status="UP">
  <hostname>216.180.251.178:27666</hostname>
  <name>www.eXg-clan.com [PRI]</name>
  <gametype>emz</gametype>
  <map>game/mp/d3refrag</map>
  <numplayers>2</numplayers>
  <maxplayers>8</maxplayers>
  <ping>34</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="70.145.26.10:50800" status="UP">
  <hostname>70.145.26.10:50800</hostname>
  <name>LAN PARTY II</name>
  <gametype>lms3</gametype>
  <map>game/enpro</map>
  <numplayers>1</numplayers>
  <maxplayers>4</maxplayers>
  <ping>69</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="68.42.196.9:27666" status="UP">
  <hostname>68.42.196.9:27666</hostname>
  <name>CoZmicShReddeR.com Clan_C7 NO_CLAN_TAGS</name>
  <gametype>lms3</gametype>
  <map>game/phobos1</map>
  <numplayers>0</numplayers>
  <maxplayers>6</maxplayers>
  <ping>40</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="69.13.26.28:27666" status="UP">
  <hostname>69.13.26.28:27666</hostname>
  <name>Nonameservers</name>
  <gametype></gametype>
  <map>game/mp/d3dm1</map>

```

```

        <numplayers>2</numplayers>
        <maxplayers>16</maxplayers>
        <ping>29</ping>
        <retries>0</retries>
</server>
<server type="DM3S" address="68.3.81.121:27666" status="UP">
  <hostname>68.3.81.121:27666</hostname>
  <name>NEW PLAYER</name>
  <gametype></gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>4</numplayers>
  <maxplayers>4</maxplayers>
  <ping>132</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="62.2.27.241:27300" status="UP">
  <hostname>62.2.27.241:27300</hostname>
  <name>SEA-Clan.Org EMZ 1.21</name>
  <gametype>emz</gametype>
  <map>game/mp/methdml</map>
  <numplayers>6</numplayers>
  <maxplayers>8</maxplayers>
  <ping>124</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="81.27.73.100:27800" status="UP">
  <hostname>81.27.73.100:27800</hostname>
  <name>the-emz.com Official EMZ v.1.30 (Beta 3) - lazeryattack.com</name>
  <gametype>emz</gametype>
  <map>swd3dml</map>
  <numplayers>3</numplayers>
  <maxplayers>4</maxplayers>
  <ping>80</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="72.226.230.127:27666" status="UP">
  <hostname>72.226.230.127:27666</hostname>
  <name>Hobbes LMS 3.0 Co-op Server</name>
  <gametype>lms3</gametype>
  <map>game/erebus3</map>
  <numplayers>3</numplayers>
  <maxplayers>4</maxplayers>
  <ping>31</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="88.149.151.205:27666" status="UP">
  <hostname>88.149.151.205:27666</hostname>
  <name>NGI [EMZ 1.21] Doom 3 Generic #1</name>
  <gametype>emz</gametype>
  <map>swd3dml</map>
  <numplayers>2</numplayers>
  <maxplayers>4</maxplayers>
  <ping>105</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="85.214.55.250:27666" status="UP">
  <hostname>85.214.55.250:27666</hostname>
  <name>Dim-ension.de-OpenCoop#2</name>
  <gametype>opencoop</gametype>
  <map>game/mars_city2</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>146</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.71.180.29:27671" status="UP">
  <hostname>64.71.180.29:27671</hostname>
  <name>Last Stronghold (CTF)</name>
  <gametype>d3xp</gametype>
  <map>game/mp/d3ctf1</map>
  <numplayers>0</numplayers>

```

```

        <maxplayers>8</maxplayers>
        <ping>87</ping>
        <retries>0</retries>
</server>
<server type="DM3S" address="12.32.216.178:27666" status="UP">
  <hostname>12.32.216.178:27666</hostname>
  <name>A2K6.COM - Doom3</name>
  <gametype></gametype>
  <map>game/mp/d3dm5</map>
  <numplayers>0</numplayers>
  <maxplayers>10</maxplayers>
  <ping>23</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.34.200.229:27668" status="UP">
  <hostname>64.34.200.229:27668</hostname>
  <name>FragArcade.com - Live Stats - #692</name>
  <gametype></gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>3</numplayers>
  <maxplayers>10</maxplayers>
  <ping>22</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="81.169.167.62:27666" status="UP">
  <hostname>81.169.167.62:27666</hostname>
  <name>Dim-ension.de-OpenCoop#1</name>
  <gametype>opencoop</gametype>
  <map>game/comml</map>
  <numplayers>1</numplayers>
  <maxplayers>4</maxplayers>
  <ping>137</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="24.129.21.245:62930" status="UP">
  <hostname>24.129.21.245:62930</hostname>
  <name>Meet Hell</name>
  <gametype></gametype>
  <map>game/mp/d3refrag</map>
  <numplayers>0</numplayers>
  <maxplayers>6</maxplayers>
  <ping>53</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="24.253.71.128:27666" status="UP">
  <hostname>24.253.71.128:27666</hostname>
  <name>VEGAS DOOM</name>
  <gametype>d3xp</gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>0</numplayers>
  <maxplayers>6</maxplayers>
  <ping>98</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="69.65.15.138:27666" status="UP">
  <hostname>69.65.15.138:27666</hostname>
  <name>[GUI] CLAN PURE SERVER</name>
  <gametype>d3xp</gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>3</numplayers>
  <maxplayers>8</maxplayers>
  <ping>23</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="71.120.142.102:27666" status="UP">
  <hostname>71.120.142.102:27666</hostname>
  <name>bustin bill</name>
  <gametype></gametype>
  <map>game/mp/d3dm2</map>
  <numplayers>1</numplayers>
  <maxplayers>4</maxplayers>

```

```

        <ping>88</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="84.9.75.218:27666" status="UP">
        <hostname>84.9.75.218:27666</hostname>
        <name>DOOM Server</name>
        <gametype></gametype>
        <map>game/mp/d3dm4</map>
        <numplayers>4</numplayers>
        <maxplayers>4</maxplayers>
        <ping>152</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="64.92.172.58:27666" status="UP">
        <hostname>64.92.172.58:27666</hostname>
        <name>HBB-clan.com</name>
        <gametype>lms3</gametype>
        <map>game/hell1</map>
        <numplayers>0</numplayers>
        <maxplayers>8</maxplayers>
        <ping>53</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="213.106.96.225:50200" status="UP">
        <hostname>213.106.96.225:50200</hostname>
        <name>CARAZEE</name>
        <gametype>lms3</gametype>
        <map>game/alphalabs1</map>
        <numplayers>3</numplayers>
        <maxplayers>4</maxplayers>
        <ping>105</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="74.96.82.43:27666" status="UP">
        <hostname>74.96.82.43:27666</hostname>
        <name>Warlord's Arena II</name>
        <gametype></gametype>
        <map>game/mp/d3dm3</map>
        <numplayers>0</numplayers>
        <maxplayers>4</maxplayers>
        <ping>43</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="84.242.67.92:27666" status="UP">
        <hostname>84.242.67.92:27666</hostname>
        <name>DooM LMS 3.0 Coop Server</name>
        <gametype>lms3</gametype>
        <map>game/caverns1</map>
        <numplayers>3</numplayers>
        <maxplayers>4</maxplayers>
        <ping>131</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="72.36.133.230:27960" status="UP">
        <hostname>72.36.133.230:27960</hostname>
        <name>Smokes Chronicles!!</name>
        <gametype></gametype>
        <map>dc/mp/dc_hopeless_alpha</map>
        <numplayers>3</numplayers>
        <maxplayers>4</maxplayers>
        <ping>54</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="62.4.81.246:30100" status="UP">
        <hostname>62.4.81.246:30100</hostname>
        <name>Doom3Maps.org - EMZ 1.21 - Server 1 by Stormix</name>
        <gametype>emz</gametype>
        <map>swd3dm2</map>
        <numplayers>0</numplayers>
        <maxplayers>4</maxplayers>
        <ping>102</ping>

```

```

        <retries>0</retries>
</server>
<server type="DM3S" address="64.182.169.193:27960" status="UP">
  <hostname>64.182.169.193:27960</hostname>
  <name>pineapple happynewyear!</name>
  <gametype></gametype>
  <map>game/steveodm2/steveodm2_lo</map>
  <numplayers>8</numplayers>
  <maxplayers>8</maxplayers>
  <ping>28</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="62.4.81.252:30700" status="UP">
  <hostname>62.4.81.252:30700</hostname>
  <name>Stormix.de Doom 3 Testserver</name>
  <gametype></gametype>
  <map>swd3dm2</map>
  <numplayers>3</numplayers>
  <maxplayers>4</maxplayers>
  <ping>101</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="69.225.52.201:27666" status="UP">
  <hostname>69.225.52.201:27666</hostname>
  <name>Dept.ofCorrections</name>
  <gametype></gametype>
  <map>game/mp/d3dm5</map>
  <numplayers>4</numplayers>
  <maxplayers>8</maxplayers>
  <ping>103</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="68.232.163.54:27666" status="UP">
  <hostname>68.232.163.54:27666</hostname>
  <name>-\\BAD Company//-</name>
  <gametype>emz</gametype>
  <map>aerod3</map>
  <numplayers>0</numplayers>
  <maxplayers>8</maxplayers>
  <ping>45</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.71.180.29:27667" status="UP">
  <hostname>64.71.180.29:27667</hostname>
  <name>Alienware.com - Server III</name>
  <gametype></gametype>
  <map>game/mp/d3dm1</map>
  <numplayers>0</numplayers>
  <maxplayers>8</maxplayers>
  <ping>85</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="194.65.14.105:27666" status="UP">
  <hostname>194.65.14.105:27666</hostname>
  <name>[ SAPOPlay | WAR-ZONE ] Tourney #1</name>
  <gametype></gametype>
  <map>game/mp/d3dm3</map>
  <numplayers>3</numplayers>
  <maxplayers>4</maxplayers>
  <ping>138</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="24.185.116.227:27666" status="UP">
  <hostname>24.185.116.227:27666</hostname>
  <name>NO - SHUTDOWN</name>
  <gametype></gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>3</numplayers>
  <maxplayers>8</maxplayers>
  <ping>18</ping>
  <retries>0</retries>

```



```

</server>
<server type="DM3S" address="195.34.7.33:27666" status="UP">
  <hostname>195.34.7.33:27666</hostname>
  <name>StreamArena Doom3 server</name>
  <gametype></gametype>
  <map>game/mp/d3dm3</map>
  <numplayers>0</numplayers>
  <maxplayers>12</maxplayers>
  <ping>136</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="82.34.85.67:27666" status="UP">
  <hostname>82.34.85.67:27666</hostname>
  <name>yo      spoony!</name>
  <gametype>lms3</gametype>
  <map>game/monorail</map>
  <numplayers>2</numplayers>
  <maxplayers>4</maxplayers>
  <ping>112</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="76.171.39.137:27666" status="UP">
  <hostname>76.171.39.137:27666</hostname>
  <name>jrsvtrew</name>
  <gametype>d3xp</gametype>
  <map>game/mp/d3ctf3</map>
  <numplayers>2</numplayers>
  <maxplayers>4</maxplayers>
  <ping>124</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="68.161.30.72:27666" status="UP">
  <hostname>68.161.30.72:27666</hostname>
  <name>Marty's World Suck</name>
  <gametype>opencoop</gametype>
  <map>game/comm1</map>
  <numplayers>1</numplayers>
  <maxplayers>4</maxplayers>
  <ping>56</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.34.200.229:27666" status="UP">
  <hostname>64.34.200.229:27666</hostname>
  <name>FragArcade.com - Live Stats - #692</name>
  <gametype></gametype>
  <map>game/mp/d3dm2</map>
  <numplayers>6</numplayers>
  <maxplayers>10</maxplayers>
  <ping>22</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="85.86.41.54:60235" status="UP">
  <hostname>85.86.41.54:60235</hostname>
  <name>Mega Ciber Zarautz 15</name>
  <gametype></gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>3</numplayers>
  <maxplayers>4</maxplayers>
  <ping>186</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.34.200.229:27667" status="UP">
  <hostname>64.34.200.229:27667</hostname>
  <name>FragArcade.com - Fragit - #800</name>
  <gametype></gametype>
  <map>game/mp/d3dm3</map>
  <numplayers>10</numplayers>
  <maxplayers>10</maxplayers>
  <ping>101</ping>
  <retries>0</retries>
</server>

```

```

<server type="DM3S" address="71.136.38.124:27666" status="UP">
  <hostname>71.136.38.124:27666</hostname>
  <name>OMG...!!!</name>
  <gametype></gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>4</numplayers>
  <maxplayers>4</maxplayers>
  <ping>103</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="71.196.171.116:27666" status="UP">
  <hostname>71.196.171.116:27666</hostname>
  <name>Spunky's LMS 3.0 USA server 1</name>
  <gametype>lms3</gametype>
  <map>game/samuel/kaiser_ld3</map>
  <numplayers>0</numplayers>
  <maxplayers>5</maxplayers>
  <ping>82</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="130.88.189.64:27666" status="UP">
  <hostname>130.88.189.64:27666</hostname>
  <name>Man Uni: Doom 3</name>
  <gametype>base</gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>4</numplayers>
  <maxplayers>4</maxplayers>
  <ping>108</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="81.169.141.133:27666" status="UP">
  <hostname>81.169.141.133:27666</hostname>
  <name>Doom 3</name>
  <gametype></gametype>
  <map>game/mp/d3dm1</map>
  <numplayers>2</numplayers>
  <maxplayers>4</maxplayers>
  <ping>145</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="58.138.60.106:27666" status="UP">
  <hostname>58.138.60.106:27666</hostname>
  <name>Hell[JAPAN]Linux</name>
  <gametype>lms3</gametype>
  <map>game/alphalabs1</map>
  <numplayers>3</numplayers>
  <maxplayers>4</maxplayers>
  <ping>315</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="84.7.2.118:27666" status="UP">
  <hostname>84.7.2.118:27666</hostname>
  <name>DOOM Server</name>
  <gametype></gametype>
  <map>game/mp/d3dm2</map>
  <numplayers>1</numplayers>
  <maxplayers>4</maxplayers>
  <ping>198</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="70.109.142.228:50139" status="UP">
  <hostname>70.109.142.228:50139</hostname>
  <name>DOOM Server</name>
  <gametype></gametype>
  <map>game/mp/d3dm2</map>
  <numplayers>3</numplayers>
  <maxplayers>4</maxplayers>
  <ping>41</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="68.229.152.194:27666" status="UP">

```

```

        <hostname>68.229.152.194:27666</hostname>
        <name>Leger.Serve</name>
        <gametype></gametype>
        <map>game/mp/d3dm3</map>
        <numplayers>2</numplayers>
        <maxplayers>2</maxplayers>
        <ping>247</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="82.47.147.14:27666" status="UP">
        <hostname>82.47.147.14:27666</hostname>
        <name>AY DOOM</name>
        <gametype></gametype>
        <map>game/mp/d3dm1</map>
        <numplayers>2</numplayers>
        <maxplayers>4</maxplayers>
        <ping>95</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="80.121.116.104:27666" status="UP">
        <hostname>80.121.116.104:27666</hostname>
        <name>Lex</name>
        <gametype>opencoop</gametype>
        <map>game/mars_city1</map>
        <numplayers>2</numplayers>
        <maxplayers>2</maxplayers>
        <ping>137</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="194.22.189.59:27666" status="UP">
        <hostname>194.22.189.59:27666</hostname>
        <name>Fagz</name>
        <gametype>opencoop</gametype>
        <map>game/caverns2</map>
        <numplayers>2</numplayers>
        <maxplayers>4</maxplayers>
        <ping>107</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="76.171.209.98:27666" status="UP">
        <hostname>76.171.209.98:27666</hostname>
        <name>QnDfix LMScoop! </name>
        <gametype>lms3</gametype>
        <map>game/alphalabs3</map>
        <numplayers>3</numplayers>
        <maxplayers>6</maxplayers>
        <ping>196</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="72.202.140.209:27666" status="UP">
        <hostname>72.202.140.209:27666</hostname>
        <name>DEATHMATCH PURE SERVER</name>
        <gametype></gametype>
        <map>game/mp/d3dm1</map>
        <numplayers>4</numplayers>
        <maxplayers>8</maxplayers>
        <ping>70</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="85.140.137.24:27666" status="UP">
        <hostname>85.140.137.24:27666</hostname>
        <name>DOOM Server</name>
        <gametype></gametype>
        <map>game/mp/d3dm2</map>
        <numplayers>3</numplayers>
        <maxplayers>3</maxplayers>
        <ping>188</ping>
        <retries>0</retries>
    </server>
    <server type="DM3S" address="24.181.252.67:27666" status="UP">
        <hostname>24.181.252.67:27666</hostname>

```

```

        <name>DOOM Server</name>
        <gametype></gametype>
        <map>swd3dm2</map>
        <numplayers>1</numplayers>
        <maxplayers>4</maxplayers>
        <ping>67</ping>
        <retries>0</retries>
</server>
<server type="DM3S" address="86.122.226.161:27666" status="UP">
    <hostname>86.122.226.161:27666</hostname>
    <name>WARP</name>
    <gametype></gametype>
    <map>game/mp/d3dm2</map>
    <numplayers>2</numplayers>
    <maxplayers>4</maxplayers>
    <ping>150</ping>
    <retries>0</retries>
</server>
<server type="DM3S" address="82.32.82.139:27666" status="UP">
    <hostname>82.32.82.139:27666</hostname>
    <name>DOOM Server</name>
    <gametype></gametype>
    <map>game/mp/d3dm1</map>
    <numplayers>1</numplayers>
    <maxplayers>4</maxplayers>
    <ping>119</ping>
    <retries>0</retries>
</server>
<server type="DM3S" address="89.35.185.16:27666" status="UP">
    <hostname>89.35.185.16:27666</hostname>
    <name>DOOM Server</name>
    <gametype></gametype>
    <map>game/mp/d3dm3</map>
    <numplayers>4</numplayers>
    <maxplayers>4</maxplayers>
    <ping>128</ping>
    <retries>0</retries>
</server>
<server type="DM3S" address="222.191.248.210:2927" status="UP">
    <hostname>222.191.248.210:2927</hostname>
    <name>DOOM Server</name>
    <gametype></gametype>
    <map>game/mp/d3dm3</map>
    <numplayers>3</numplayers>
    <maxplayers>4</maxplayers>
    <ping>390</ping>
    <retries>0</retries>
</server>
<server type="DM3S" address="62.4.81.249:30000" status="UP">
    <hostname>62.4.81.249:30000</hostname>
    <name>Doom3Maps.org - EMZ 1.21 - Server 3 by Stormix</name>
    <gametype>emz</gametype>
    <map>swd3dm2</map>
    <numplayers>0</numplayers>
    <maxplayers>4</maxplayers>
    <ping>101</ping>
    <retries>0</retries>
</server>
<server type="DM3S" address="70.86.179.2:27666" status="UP">
    <hostname>70.86.179.2:27666</hostname>
    <name>Alienware.com - Server I</name>
    <gametype></gametype>
    <map>game/mp/d3dm4</map>
    <numplayers>5</numplayers>
    <maxplayers>8</maxplayers>
    <ping>57</ping>
    <retries>0</retries>
</server>
<server type="DM3S" address="217.72.215.70:27666" status="UP">
    <hostname>217.72.215.70:27666</hostname>
    <name>$H(+)Ts.Clan Emz v1.21 </name>

```

```

    <gametype>emz</gametype>
    <map>swd3dm1</map>
    <numplayers>5</numplayers>
    <maxplayers>5</maxplayers>
    <ping>131</ping>
    <retries>0</retries>
</server>
<server type="DM3S" address="74.96.82.43:27668" status="UP">
  <hostname>74.96.82.43:27668</hostname>
  <name>Warlord's Arena</name>
  <gametype></gametype>
  <map>game/mp/d3dm1</map>
  <numplayers>3</numplayers>
  <maxplayers>4</maxplayers>
  <ping>42</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="200.171.26.133:39883" status="UP">
  <hostname>200.171.26.133:39883</hostname>
  <name>!!CYBERMAC BRAZIL!!</name>
  <gametype></gametype>
  <map>game/mp/d3dm3</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>170</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.71.180.29:27666" status="UP">
  <hostname>64.71.180.29:27666</hostname>
  <name>Slaughter House Dm...</name>
  <gametype></gametype>
  <map>aerod3</map>
  <numplayers>0</numplayers>
  <maxplayers>8</maxplayers>
  <ping>85</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="70.86.179.3:27666" status="UP">
  <hostname>70.86.179.3:27666</hostname>
  <name>Alienware.com - Server II</name>
  <gametype></gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>7</numplayers>
  <maxplayers>8</maxplayers>
  <ping>51</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="65.98.61.18:27666" status="UP">
  <hostname>65.98.61.18:27666</hostname>
  <name>Generation {XS} EMZ Server</name>
  <gametype>emz</gametype>
  <map>game/mp/gwdm1</map>
  <numplayers>0</numplayers>
  <maxplayers>6</maxplayers>
  <ping>16</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.71.180.29:27670" status="UP">
  <hostname>64.71.180.29:27670</hostname>
  <name>Slaughter House Tourney..</name>
  <gametype></gametype>
  <map>game/mp/d3dm4</map>
  <numplayers>0</numplayers>
  <maxplayers>8</maxplayers>
  <ping>86</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="195.4.104.156:27666" status="UP">
  <hostname>195.4.104.156:27666</hostname>
  <name>gzdoom</name>
  <gametype></gametype>

```

```

        <map>game/mp/d3dm1</map>
        <numplayers>0</numplayers>
        <maxplayers>4</maxplayers>
        <ping>110</ping>
        <retries>0</retries>
</server>
<server type="DM3S" address="128.174.157.33:27666" status="UP">
  <hostname>128.174.157.33:27666</hostname>
  <name>BenZone Doom3 CoOp</name>
  <gametype>lms3</gametype>
  <map>game/mc_underground</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>37</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="85.14.217.81:30000" status="UP">
  <hostname>85.14.217.81:30000</hostname>
  <name>TsaiTschen Q4</name>
  <gametype>q4base</gametype>
  <map>mp/q4dm5</map>
  <numplayers>0</numplayers>
  <maxplayers>12</maxplayers>
  <ping>96</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="216.21.45.37:27666" status="UP">
  <hostname>216.21.45.37:27666</hostname>
  <name>Hells Kitchen Chicago</name>
  <gametype>lms3</gametype>
  <map>game/delta2a</map>
  <numplayers>3</numplayers>
  <maxplayers>6</maxplayers>
  <ping>36</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="216.52.148.125:27666" status="UP">
  <hostname>216.52.148.125:27666</hostname>
  <name>s3lf Pvt Tourney (chic)</name>
  <gametype></gametype>
  <map>swd3dm2</map>
  <numplayers>0</numplayers>
  <maxplayers>4</maxplayers>
  <ping>25</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="64.71.180.29:27669" status="UP">
  <hostname>64.71.180.29:27669</hostname>
  <name>Slaughter House Team Dm...</name>
  <gametype></gametype>
  <map>swd3dm1</map>
  <numplayers>0</numplayers>
  <maxplayers>8</maxplayers>
  <ping>86</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="85.0.118.148:11366" status="UP">
  <hostname>85.0.118.148:11366</hostname>
  <name>DOOM Server</name>
  <gametype>opencoop</gametype>
  <map>game/cpuboss</map>
  <numplayers>2</numplayers>
  <maxplayers>4</maxplayers>
  <ping>137</ping>
  <retries>0</retries>
</server>
<server type="DM3S" address="195.13.62.54:28100" status="TIMEOUT">
  <hostname>195.13.62.54:28100</hostname>
</server>
<server type="DM3S" address="83.5.252.46:27666" status="TIMEOUT">
  <hostname>83.5.252.46:27666</hostname>

```

```

</server>
<server type="DM3S" address="74.64.88.101:27666" status="TIMEOUT">
  <hostname>74.64.88.101:27666</hostname>
</server>
<server type="DM3S" address="199.222.140.96:27666" status="TIMEOUT">
  <hostname>199.222.140.96:27666</hostname>
</server>
<server type="DM3S" address="218.16.242.6:27666" status="TIMEOUT">
  <hostname>218.16.242.6:27666</hostname>
</server>
<server type="DM3S" address="87.217.12.94:11635" status="TIMEOUT">
  <hostname>87.217.12.94:11635</hostname>
</server>
<server type="DM3S" address="200.9.166.185:27666" status="TIMEOUT">
  <hostname>200.9.166.185:27666</hostname>
</server>
<server type="DM3S" address="74.60.25.231:1348" status="TIMEOUT">
  <hostname>74.60.25.231:1348</hostname>
</server>
<server type="DM3S" address="87.110.7.235:27666" status="TIMEOUT">
  <hostname>87.110.7.235:27666</hostname>
</server>
<server type="DM3S" address="217.224.194.7:1451" status="TIMEOUT">
  <hostname>217.224.194.7:1451</hostname>
</server>
<server type="DM3S" address="24.181.254.88:1353" status="TIMEOUT">
  <hostname>24.181.254.88:1353</hostname>
</server>
<server type="DM3S" address="190.49.12.106:62056" status="TIMEOUT">
  <hostname>190.49.12.106:62056</hostname>
</server>
<server type="DM3S" address="65.184.191.139:27666" status="TIMEOUT">
  <hostname>65.184.191.139:27666</hostname>
</server>
<server type="DM3S" address="74.135.107.73:16234" status="TIMEOUT">
  <hostname>74.135.107.73:16234</hostname>
</server>
<server type="DM3S" address="217.224.194.7:1452" status="TIMEOUT">
  <hostname>217.224.194.7:1452</hostname>
</server>
<server type="DM3S" address="190.49.12.106:62042" status="TIMEOUT">
  <hostname>190.49.12.106:62042</hostname>
</server>
<server type="DM3S" address="217.147.80.89:27667" status="TIMEOUT">
  <hostname>217.147.80.89:27667</hostname>
</server>
</qstat>

```

A6.2 Single Server Data

Query sample #1

```

<?xml version="1.0" encoding="iso-8859-1"?>
<qstat>
  <server type="DM3S" address="195.34.7.33:27666" status="UP">
    <hostname>195.34.7.33:27666</hostname>
    <name>StreamArena Doom3 server</name>
    <gametype></gametype>
    <map>game/mp/d3dm4</map>
    <numplayers>0</numplayers>
    <maxplayers>12</maxplayers>
    <ping>588</ping>
    <retries>0</retries>
    <rules>
      <rule name="protocol">1.40</rule>
      <rule name="sv_punkbuster">1</rule>
      <rule name="si_version">DOOM 1.3.1302 linux-x86 May 12 2005
15:04:27</rule>
      <rule name="fs_game_base"></rule>
      <rule name="fs_game"></rule>

```

```

        <rule name="si_idleServer">0</rule>
        <rule name="net_serverDedicated">1</rule>
        <rule name="si_maxPlayers">12</rule>
        <rule name="si_serverURL"></rule>
        <rule name="si_spectators">1</rule>
        <rule name="si_usepass">0</rule>
        <rule name="si_warmup">0</rule>
        <rule name="si_teamDamage">0</rule>
        <rule name="si_timeLimit">40</rule>
        <rule name="si_fragLimit">30</rule>
        <rule name="si_map">game/mp/d3dm4</rule>
        <rule name="si_gameType">deathmatch</rule>
        <rule name="si_name">StreamArena Doom3 server</rule>
        <rule name="si_pure">1</rule>
        <rule name="gamename">baseDOOM-1</rule>
        <rule name="osmask">0x7</rule>
    </rules>
    <players>
    </players>
</server>
</qstat>

```

Query Sample #2

```

<?xml version="1.0" encoding="iso-8859-1"?>
<qstat>
  <server type="DM3S" address="195.34.7.33:27666" status="UP">
    <hostname>195.34.7.33:27666</hostname>
    <name>StreamArena Doom3 server</name>
    <gametype></gametype>
    <map>game/mp/d3dm4</map>
    <numplayers>0</numplayers>
    <maxplayers>12</maxplayers>
    <ping>134</ping>
    <retries>0</retries>
    <rules>
      <rule name="protocol">1.40</rule>
      <rule name="sv_punkbuster">1</rule>
      <rule name="si_version">DOOM 1.3.1302 linux-x86 May 12 2005
15:04:27</rule>
      <rule name="fs_game_base"></rule>
      <rule name="fs_game"></rule>
      <rule name="si_idleServer">0</rule>
      <rule name="net_serverDedicated">1</rule>
      <rule name="si_maxPlayers">12</rule>
      <rule name="si_serverURL"></rule>
      <rule name="si_spectators">1</rule>
      <rule name="si_usepass">0</rule>
      <rule name="si_warmup">0</rule>
      <rule name="si_teamDamage">0</rule>
      <rule name="si_timeLimit">40</rule>
      <rule name="si_fragLimit">30</rule>
      <rule name="si_map">game/mp/d3dm4</rule>
      <rule name="si_gameType">deathmatch</rule>
      <rule name="si_name">StreamArena Doom3 server</rule>
      <rule name="si_pure">1</rule>
      <rule name="gamename">baseDOOM-1</rule>
      <rule name="osmask">0x7</rule>
    </rules>
    <players>
    </players>
  </server>
</qstat>

```