Project number: MLC-NG03

The Effects of Packet Loss and Latency on Player Performance in Unreal Tournament 2003

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:

Thomas P. Beigbeder

Rory J. Coughlan

Corey C. Lusher

John J. Plunkett

Date: March 3, 2004

Approved:

Professor Mark L. Claypool

Professor Emmanuel O. Agu

Abstract

First Person Shooter computer games are highly sensitive to network fluctuations due to the need for precise movement and aiming. In this project we studied player performance in Unreal Tournament 2003 under varying amounts of latency and packet loss. After breaking the game down into specific components and running tests under controlled network conditions we determined that packet loss and latency levels likely to be encountered in real world settings will not drastically impact player performance.

Table of Contents

1. Introduction 5
2. Related Work 8
3. Background 10
4. Technologies 15
5. Overview 17
6. Traffic Analysis 19
7. Movement Tests 26
   7.1 Simple Movement 26
      7.1.1 Packet Loss 27
      7.1.2 Latency 27
      7.1.3 Summary 27
   7.2 Complex Movement 28
      7.2.1 Packet Loss 31
      7.2.2 Latency 32
      7.2.3 Summary 33
8. Precise Shooting Tests 34
   8.1 Packet Loss 34
   8.2 Latency 36
   8.3 Summary 37
9. Restricted Deathmatch Test 38
   9.1 Packet Loss 38
   9.2 Latency 39
   9.3 Summary 40
10. Full Deathmatch Test 41
   10.1 Packet Loss 41
   10.2 Latency 42
   10.3 Summary 43
11. Subjective Quality Assessment 44
12. Conclusions 46
13. Future Work 49
14. References 50
A. Data Tables 51
   A.1 Complex Movement Test Data 51
      A.1.1 Complex Movement with Packet Loss 51
      A.1.2 Complex Movement with Latency 51
   A.2 Precise Shooting Data 52
      A.2.1 Precise Shooting with Packet Loss 52
      A.2.2 Precise Shooting with Latency 53
   A.3 Restricted Deathmatch Data 54
      A.3.1 Packet Loss 54
      A.3.2 Latency 55
   A.4 Full Game Data 56
      A.4.1 Full Game with Packet Loss 56
      A.4.2 Full Game with Latency 57

Table of Figures

Figure 1: UT2003 Scoreboard 10
Figure 2: Firing the Rocket Launcher 11
Figure 3: Weapon Types 12
Figure 4: Fully Zoomed Lightning Gun 13
Figure 5: Testbed Setup 15
Figure 6: Game Server Latency Distribution 19
Figure 7: Game Server Variance Distribution 20
Figure 8: Game Server Packet Loss Distribution 21
Figure 9: Cumulative Density Function of the Packet Sizes 22
Figure 10: Bitrate vs Time (no Packet Loss) 23
Figure 11: Bitrate vs Time (with Packet Loss) 23
Figure 12: Average Bitrate and Standard Deviation 24
Figure 13: Packet Arrival Times (Client to Server) 24
Figure 14: Packet Arrival Times (Server to Client) 25
Figure 15: Start of Complex Movement Test 28
Figure 16: Middle of Obstacle Course 29
Figure 17: Obstacle Course Continued 30
Figure 18: End of Complex Movement Test 30
Figure 19: Complex Movement with Packet Loss 31
Figure 20: Complex Movement with Latency 32
Figure 21: Precise Shooting with Packet Loss 35
Figure 22: Precise Shooting with Latency 37
Figure 23: 5 Minute Restricted Deathmatch with Packet Loss 39
Figure 24: 5 Minute Restricted Deathmatch with Latency 40
Figure 25: 5 Minute Deathmatch with Packet Loss 42
Figure 26: 5 Minute Deathmatch with Latency 43

1. Introduction

In recent years the performance of personal computers has increased dramatically, while declining costs have made such systems available to mainstream users. With this ever growing user base, computer games and online gaming have become more popular as well. The increase in processing power has been accompanied by broadband Internet connections in homes that have higher capacities and lower latencies than traditional modems. With these hardware improvements, the potential for large profits has driven more and more game developers to incorporate multiplayer features into their products.

The Internet, however, was not designed for the real-time nature of computer games and their needs for high levels of interactivity. Despite being one of the predominant protocols for Internet traffic, the Transmission Control Protocol (TCP) is often not used for multiplayer game traffic because it requires packets to be delivered reliably and sequentially. Excessively delayed or erroneous game packets should sometimes be dropped and not retransmitted in order to reduce latency. For these reasons most game developers use the User Datagram Protocol (UDP). UDP has no retransmission or acknowledgement features built in, eliminating delays due to packet retransmissions resulting in more rapid transmissions.

While several game genres incorporate multiplayer features, the most popular are parlor games such as chess and various card games. However, parlor games are fairly tolerant of varying network conditions since most are turn-based. The most popular genres after parlor games are First Person Shooters (FPS) [LW01]. FPS games are those in which a user views the world through the eyes of a virtual character and controls their actions via a keyboard and mouse. In these games, players collect weapons and power-

ups and attempt to destroy other players. FPS games are highly sensitive to changes in delay, loss, and available capacity. The need for precise movement and aiming requires constant updates from a game server, which requires low latencies and packet loss rates. Despite increasing throughput speeds, broadband connections can still be subject to high amounts of latency and packet loss.

Some of the most popular FPS games have descended from two game lineages, using either a Quake or Unreal-based game engine [FCF+03]. According to Gamespy, the most popular FPS is Half-life and its various off-shoots such as Counter-strike, Day of Defeat, and Team Fortress Classic.[1] For this reason, games based on Half-life are frequently used when analyzing how various network factors affect game-play. Researchers however, have discovered that games using the Unreal engine, specifically Unreal Tournament 2003, have smaller overall packet sizes than do Quake based games and use less capacity, but send packets more frequently [FCF+03].

Using a game from a lineage different than Counter-strike gives us a better understanding of how network games perform. Counter-strike may also have specific strengths or weaknesses that may not appear unless compared with different games in the genre. From a scientific standpoint, testing other games will give a more accurate indication of how well various multiplayer implementations perform under varying degrees of packet loss and other network irregularities.

There are still several areas of multiplayer network gaming that have yet to be explored. First, we have not found any projects that systematically examine the effect of packet loss on user performance in FPS or other genres. We believe that packet loss may have a similar impact on user performance as does latency and therefore is an important

[1] Based on game server populations from October, 2003. http://www.gamespy.com/stats/

area of study. Such loss studies will be increasingly important as transmission media, such as wireless channels become widely adopted, since they are more prone to higher loss rates. The purpose of our MQP is to examine the effects of both latency and packet loss over a network on a FPS, and evaluate how these factors affect user performance and satisfaction.

For this project we used Unreal Tournament 2003 (UT2003) in our experiments.[2] UT2003 is currently very popular, with approximately 1700 servers and 4400 players online at any given time [Gamespy, October 2003]. UT2003 comes from the award winning Unreal series from Epic Games, and in addition to receiving excellent reviews, won the Best of Show award at the Electronic Entertainment Expo 2002[3] from such computer-game oriented World Wide Web sites as gamespot.com, ign.com, xgr.com and gamehelper.com.

In our approach to this project we divided aspects of user interaction of UT2003 into individual components in order to isolate particular facets of game-play. We focused on a player's ability to move around in the game environment, his ability to aim with a weapon that requires precision, his ability to aim with less accurate weapons, and a player's ability to both move and shoot simultaneously. We developed experiments that isolated these components of the game and then analyzed the impact of packet loss and latency on the player's performance. In order to perform these experiments we setup a testbed where we could systematically control latency and loss rates. Using this testbed and UT2003's "LAN" network connection setting, we performed tests with standard maps, weapons, mutators, and bots to test the various game components.

[2] http://www.unrealtournament.com
[3] Los Angeles, California. May, 2002.

2. Related Work

For the purposes of our project we will refer to research that has already been completed into network issues on performance of network based computer games. In addition to providing us with some context for our work, it also gave us background information about the way computer games send and receive traffic over networks.

Past work on the effects of network issues on network games has primarily been limited to research into latency (and in some cases, variance in latency, or jitter). Many papers have been written on the topic and some can be found in the ACM's digital library.[4] Some of these studies set out to determine how much latency a user would consider acceptable and how much latency would be deemed unplayable [A02]. Games from several genres have been examined, including First Person Shooter (FPS), Real-Time Strategy (RTS), racing, and simulation games [PW02]. One Counter-strike server was configured to capture statistics about players' behavior and latency over a period of seven days [FCF+02]. Others have done similar work with Counter-Strike servers and have analyzed network traffic statistics in relation to players' habits [H01]. Other studies have taken a similar approach with a Quake 3 Arena server [A01]. The general consensus of these researchers is that latency is not desirable when playing games online and players are conditioned to avoid high ping times.

In addition to papers published by researchers at other institutions, a Major Qualifying Project was completed at Worcester Polytechnic Institute in which the students studied increasing amounts of latency to measure its effects on various aspects of game play in a Real Time Strategy game [SEB03]. The students discovered that

[4] http://www.acm.org/dl

latency has a negligible effect on gameplay. Our project also focused heavily on the aspect of network latency. In our case, however, we studied its impact on a First Person Shooter.

Researchers have also looked at the network traffic itself [FCF+03]. They have examined the nature, characteristics, amount of traffic, packet sizes and distributions that have been created by game servers and clients and the aggregate bandwidth used. Studies have also compared the results of traffic analysis between games. Research involving FPS games have mostly studied the Quake lineage, most notably Counter-Strike [FCF+02]. A Major Qualifying Project was also completed at Worcester Polytechnic Institute that focused on capturing and analyzing the network traffic created by both Counter-strike and Starcraft [LW01]. The researchers investigating network traffic found that, for the most part, multiplayer computer games sent relatively small packets very frequently.

3. Background

Unreal Tournament 2003 is an online first person shooter game, in which up to 32 players can compete simultaneously on a single server. These players can compete on a variety of maps, in a variety of modes. UT2003 boasts more than 35 indoor and outdoor maps created by the game makers and packaged with the game, along with many user-created custom maps that can be acquired either from websites or simply by joining a server running a custom map. There are five multiplayer modes users can compete in: Deathmatch, Team Deathmatch, Capture the Flag, Double Domination and Bombing Run.

In Deathmatch, players compete in a free-for-all match, trying to kill as many of the opposing players as possible, while limiting the number of times they themselves are killed. At the end of the match the player with the highest score wins. Figure 1 shows a typical scoreboard from a Deathmatch game. Team Deathmatch is very similar to

Figure 1: UT2003 Scoreboard

Deathmatch. The only difference is that instead of a complete free-for-all, the players are split into two teams and the team with the highest combined score wins. Capture the Flag sets two teams against each other, in which these teams must try to both protect their own flag while trying to capture the opposing team's flag. The match ends when one team achieves a set number of captures or a time limit expires. In Double Domination, teams fight to capture and control specific key points of a map. The more of these points a team holds the more points they are awarded. Bombing Run, a futuristic football style match has teams passing and running with a ball trying to either cross a goal for 7 points, or shoot the ball into the goal for 3 points. In both Double Domination and Bombing Run the match ends when one team has achieved a set number of points, or the time limit expires.

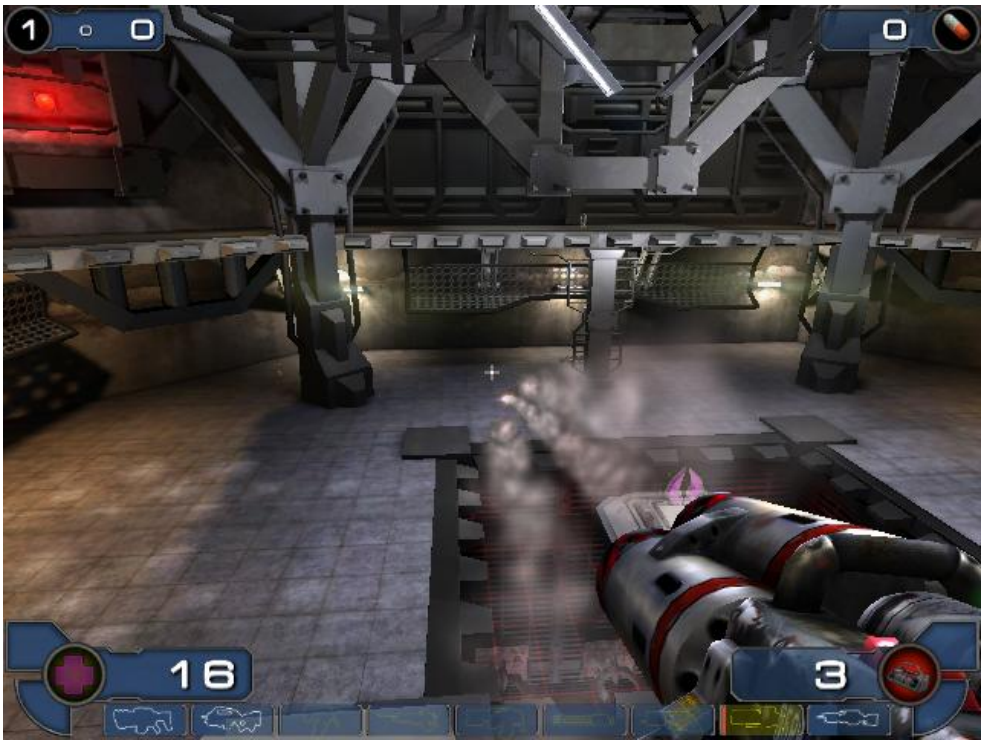

Figure 2: Firing the Rocket Launcher

In UT2003 the most popular modes of play are Deathmatch and Capture the Flag. For this reason all of our tests were performed in Deathmatch mode using Deathmatch or

Capture the Flag maps. Despite the slight differences in gameplay modes we feel all UT2003 games contain the same basic gameplay components.

Assisting players in destroying those who oppose them, UT2003 boats a large assortment of futuristic weapons. Figure 3 shows a table of weapons broken into four categories. We felt that a player's performance is directly related to their ability to hit a target. For this reason we categorized the weapons by the amount of precision that is required to use them. Higher precision weapons tend to be more difficult to use effectively when there are lost or delayed packets. We believed that the weapons that required less precision would allow a player to compensate for high amounts of latency or packet loss. For this reason we decided to limit the type of weapons available in our test games. The typical in-game weapons range from the standard machine gun to the Redeemer, which is a remote-controlled miniature rocket and the most powerful weapon in the game.

Precision Required	Weapons	Comments
High Precision	Shock Rifle, Link Gun, Lightning Gun (Figure 4)	These weapons require a player to aim with a high amount of precision in order to hit the target.
Medium Precision	Assault Rifle, Minigun, Biorifle	These weapons allow for less accurate shots to connect with their target.
Low Precision	Flak Cannon, Rocket Launcher (Figure 2), Redeemer, Ion Painter	These weapons require a player to merely aim in the vicinity of the target in order to hit it.
Other	Shield Gun, Translocator, Ball Launcher	These weapons are less commonly used, or serve special purposes within particular types of games.

Figure 3: Weapon Types

Complementing these extremes are nine more commonly used weapons. These include the Minigun, capable of firing high volumes of bullets in a very short time, the Flak Cannon, used to scatter shards of metal in the general vicinity of your opponents, and the Rocket Launcher, able to load and launch up to three rockets at a time. Along with the Lightning Gun, UT2003's version of a sniper rifle, there are many ways for players to deal with their opponents.

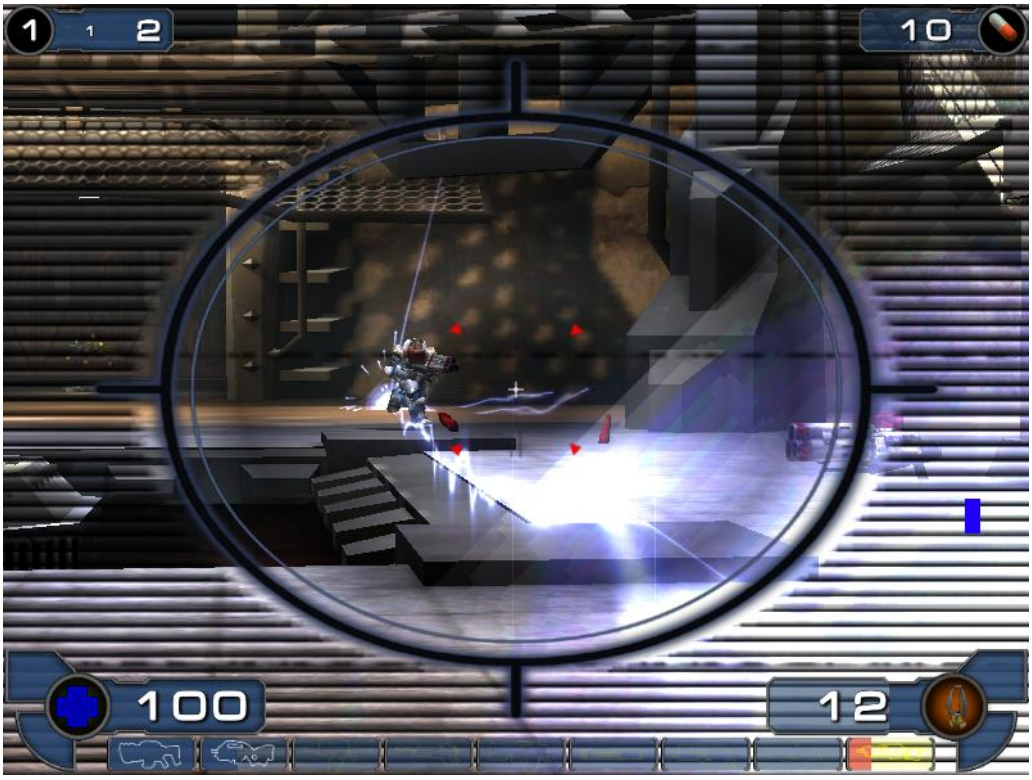

Figure 4: Fully Zoomed Lightning Gun

In addition to the numerous maps, weapons and gameplay modes, UT2003 also comes standard with two more features: bots and mutators. Bots are computer controlled players, each with their own personality and play style. Bots are used when playing UT2003 single player games or they can be used for multiplayer games when not enough human players are present. When used this way, bots are run on the game server. Mutators are custom modifications to the game environment that allow unique scenarios

to be added to a map. Some common mutators are quad-jump, allowing a player to jump 4 times in mid-air, and intsta-gib, limiting weapon choice to the Shock Rifle and making it so every shot will instantly kill your opponent.

For our experiments we had four users, encompassing three different types of players. Two users were highly experienced in both FPS games in general, and also familiar with the Unreal game lineage. One user had experience with FPS games, but was not familiar with the Unreal games. The fourth user had little prior experience with FPS games, and was considered a novice player.

4. Technologies

We performed our tests in a lab with four client computers running the latest version of Unreal Tournament 2003 (v2225). Figure 5 shows our testbed setup. The four clients were connected to a 10Mbps switch, which in turn connected to one of three network interface cards in a computer running Linux and a DHCP service. The second of these network cards connected directly to another computer running Unreal Tournament 2003 as a dedicated server. The third network card connected directly to another computer configured to act as a gateway to the WPI network and the Internet.

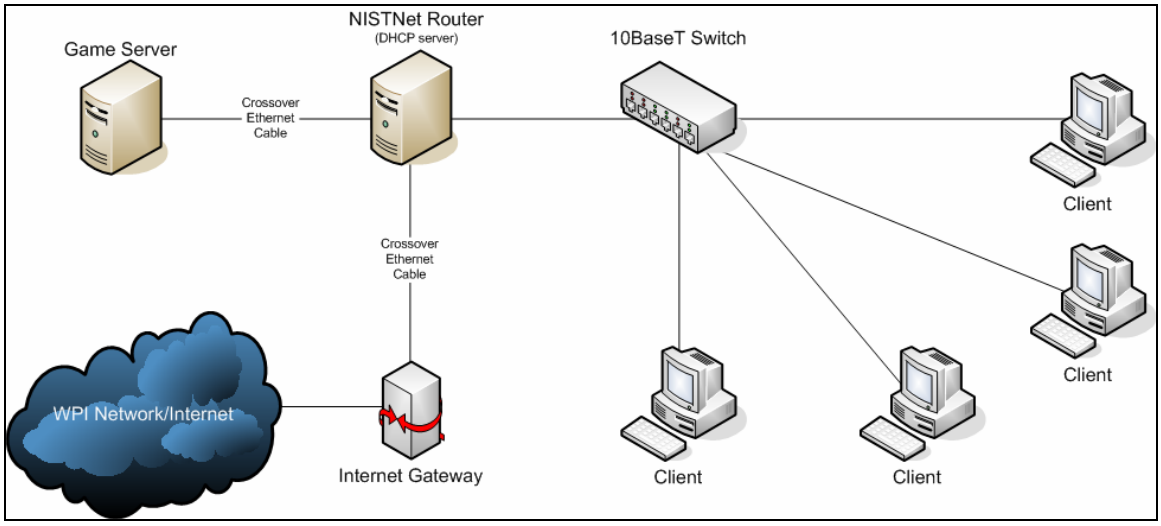

Figure 5: Testbed Setup

We also utilized two tools to control and capture network traffic. The first of these programs was NIST Net[5], which is a network emulator that was developed by the National Institute of Standards and Technology. By installing NIST Net on the router in between the server and the clients, we were able to systematically control latency and percentage of packet loss for individual clients.

[5] http://snad.ncsl.nist.gov/nistnet/

The second program we used was Ethereal[6], which is a tool that is capable of capturing all the network packets passing through a system. We ran Ethereal on the router and were able to capture all the packets traveling between the clients and server.

A third tool we used was All Seeing Eye[7], developed by UDP Soft. All Seeing Eye (ASE) is a server browser for a variety of online multiplayer computer games, including Unreal Tournament 2003. We used ASE to gather server statistics that we then used to pick levels of packet loss and latency for our tests.

[6] http://www.ethereal.com
[7] http://www.udpsoft.com/eye

5. Overview

The previous sections of this paper have presented background information about UT2003, and a summary of work done by others on related subject matter so as to provide the reader with some necessary information about multiplayer computer gaming. This section will briefly describe the type of tests we ran and what we hoped to gain by running them.

The first tests we ran were packet traces of a typical game of UT2003. We did this in order to see what the standard network traffic of a multiplayer game looked like. We wanted to see how often packets were sent both from the client to the server and from the server to the client. We also wanted information about how large the packets were, and the total bitrate being used. After capturing the packets of a standard game, we did the same analysis for games with a specific amount of induced latency, and a specific amount of induced packet loss to determine if any element of the network traffic changed as a result of packet loss or latency.

Since the majority of servers showed little or no packet loss, we never used more than 6% loss in any of our tests. Furthermore, very few servers had latencies higher than 300ms so the highest extreme we used in our latency tests was 400ms. Graphs of the cumulative distribution functions for the data gathered with ASE can be found in chapter 6 of this paper.

Our next series of tests were designed to split the game into its basic components and test them for sensitivity to packet loss and latency levels. We did this by performing controlled tests of simple movement, complex movement, precise shooting, restricted games, and unrestricted full games.

Our movement tests consisted of predefined running courses in regular game maps. The simple movement test was a straight line running test, while the complex movement was an obstacle course around a map that required complicated maneuvers.

The next tests we performed were shooting tests designed to gauge the impact of packet loss and latency on a player's ability to aim with a certain degree of precision at a moving target. This was executed by designating one player, without any added packet loss or latency, to dodge shots while another player, given different amounts of packet loss or latency and positioned far away, would try to shoot the first player with a gun that requires precise aiming.

Once we had tested the basic components of gameplay individually we started to combine them. We ran a full Deathmatch game with one player against a computer controlled bot but limited weapon choice to a gun that requires a certain degree of aiming. We then induced different levels of latency and packet loss on the player's network connection. We then ran the same test with the restriction on weapon choice removed in order to see how a real world game would be affected.

6. Traffic Analysis

This chapter describes and summarizes the results of our network traffic analysis of UT2003. Initially, we gathered server statistics to gain a better understanding of typical game playing conditions on the Internet. All of our server statistics were gathered using the All Seeing Eye tool and WPI's Internet connection as well as a DSL connection (1.5Mbit/128k) on a Sunday night at 10:00 pm EST. We also analyzed the size of packets sent during the game and calculated the game's average bitrate. Finally, we briefly look at the frequency of transmissions sent between the client and the server.

Figure 6 is a Cumulative Density Function (CDF) showing typical game server latencies. From this graph we can see that about 40% of all UT2003 game servers have latencies of less than 100ms, another 40% are between 100 and 140ms, and only 20% of all servers exhibit latencies greater than 140ms. This data provided the guideline for the levels of latency that would be used in our tests.

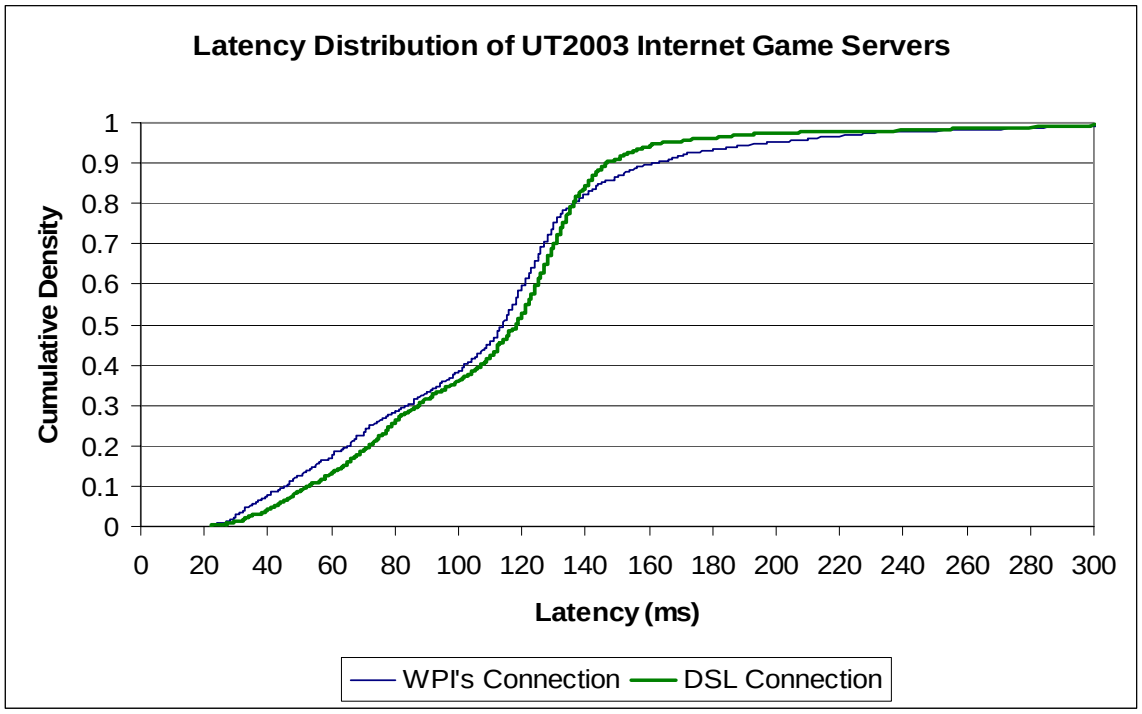

Figure 6: Game Server Latency Distribution

Figure 7 is a CDF showing the levels of ping variance (or flux) on Internet game servers. The All Seeing Eye tool measures variance by sending multiple ping packets to each game server and measuring the difference between the highest and lowest result. Interestingly, WPI's connection, a 100Mbit LAN directly connected to the Internet, seemed to have significantly more variance than that of the DSL connection we used. We repeated this test at a different time and obtained similar results.

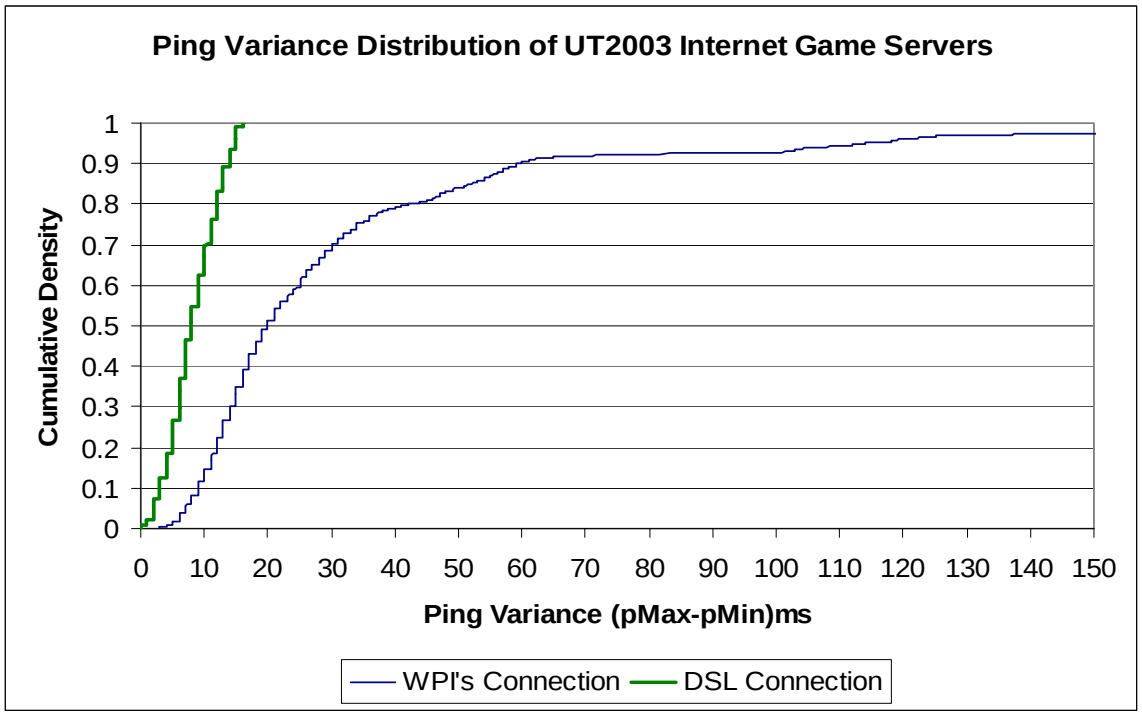

Figure 7: Game Server Variance Distribution

Figure 8 is a Complementary CDF on a logarithmic scale showing the levels of packet loss on typical game servers. As this graph shows, typical percentages of packet loss are quite low; only a few servers on WPI's connection exhibited packet loss and there was no loss on the DSL connection. However, using All Seeing Eye as a measure of packet loss is problematic for two reasons. First, since loss typically occurs in bursts, a quick scan of a server may not properly report loss rates at any given time. Second,

packet loss can also be caused by poor UT2003 server configuration. In this case, loss would not be discovered unless you were to actually join the server.

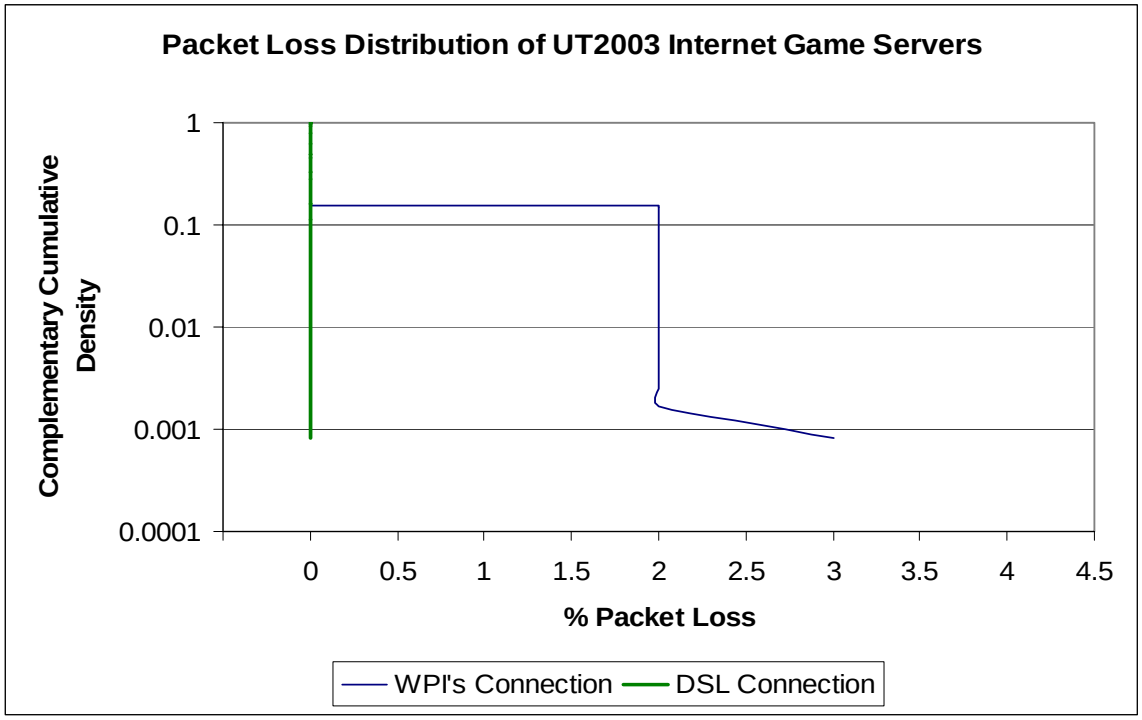

Figure 8: Game Server Packet Loss Distribution

We also gathered data in the form of packet traces. Using the tool Ethereal, we were able to capture and analyze all network traffic that occurs while playing UT2003 which we subsequently exported as a tcpdump file. Using the Ethereal captures we were able to create graphs that show the size of packets transmitted from a client to the server and back. We also determined the frequency at which packets were sent from the server to the client and vice versa. These packet traces allowed us to extrapolate how the game handles information sent between the server and client. When compared with the network traces of other FPS games the packet trace results allow us to make generalizations about FPS games that show similar network performance.

Figure 9 shows a cumulative density function of packet sizes sent during a typical game of UT2003. The size of each packet includes the IP header as well as any game data contained in the payload. For this test, we had one player matched against two bots in a small map (DM-GAEL) that is included with the game. Packets were captured for 120 seconds during the middle of the 5 minute match. We repeated this test four times under different conditions of packet loss and latency. The results are somewhat surprising, as higher levels of packet loss and latency did not appear to have a significant effect on the size of the packets. The only test that seemed to be any different was the default test with no loss and no latency. It has fewer packets of size 74 and more packets in the 76 to 90 bytes range. We repeated this test and obtained similar results.

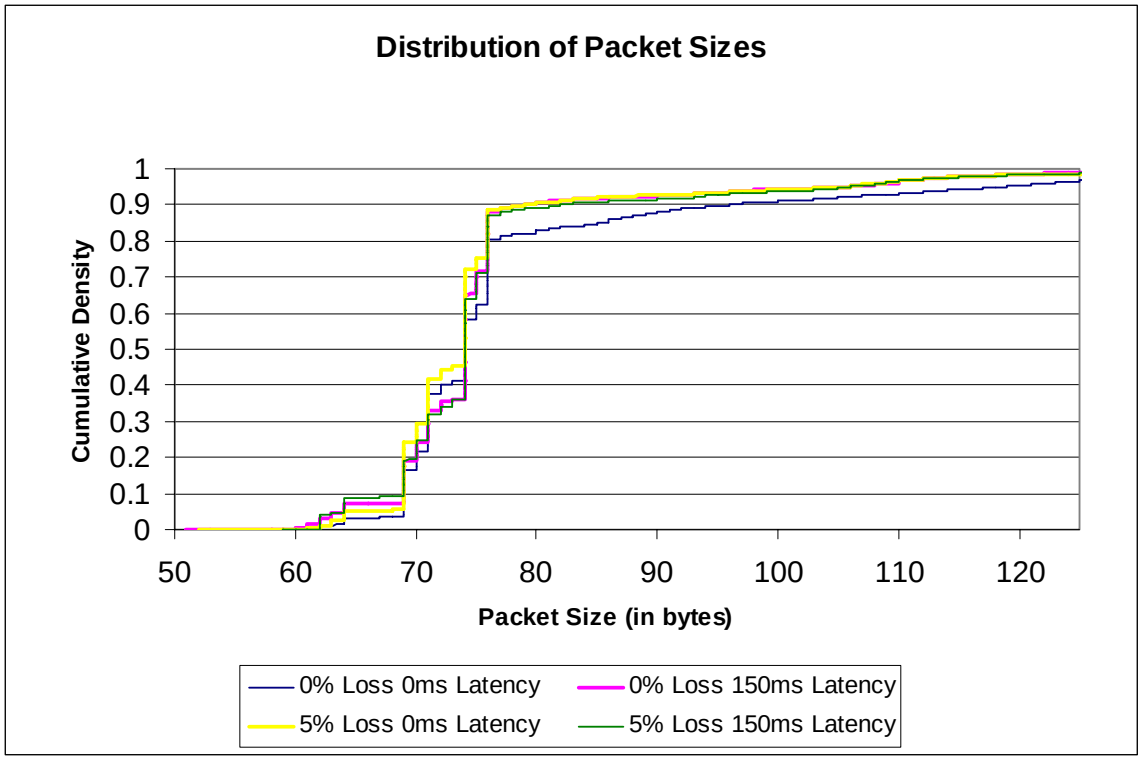

Figure 9: Cumulative Density Function of the Packet Sizes

Using the same data as in Figure 9 we were able to calculate the game's bitrate over time. Figures 10 and 11 show the bitrate we recorded, both sampled every 500ms.

From these graphs, it appears that neither packet loss nor latency has a significant effect on the game's bitrate. Figure 12 contains the game's average bitrate and the standard deviation. Once again, latency and packet loss have little effect.

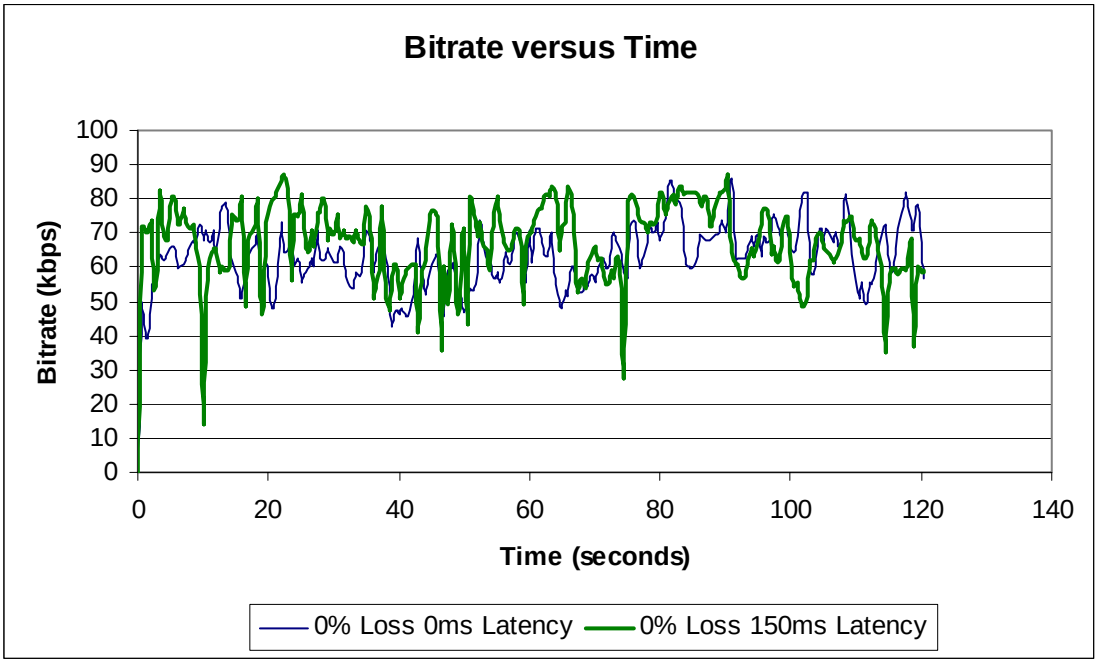

Figure 10: Bitrate vs Time (no Packet Loss)

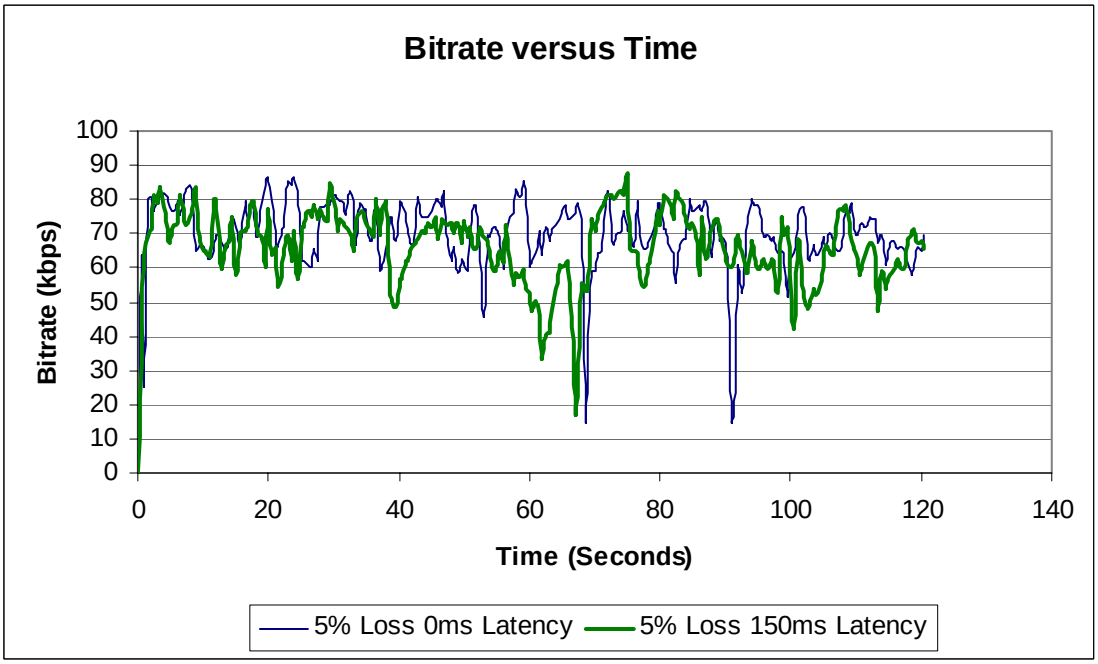

Figure 11: Bitrate vs Time (with Packet Loss)

Loss	Latency	Average Bitrate	STD
0%	0ms	63.15	9.33
0%	150ms	67.12	11.90
5%	0ms	69.87	10.86
5%	150ms	66.24	11.22

Figure 12: Average Bitrate and Standard Deviation

Finally, we analyzed the packet arrival times between the game client and the server. Packet arrival time refers to the amount of time that passes before the next packet is sent by either the client or server.

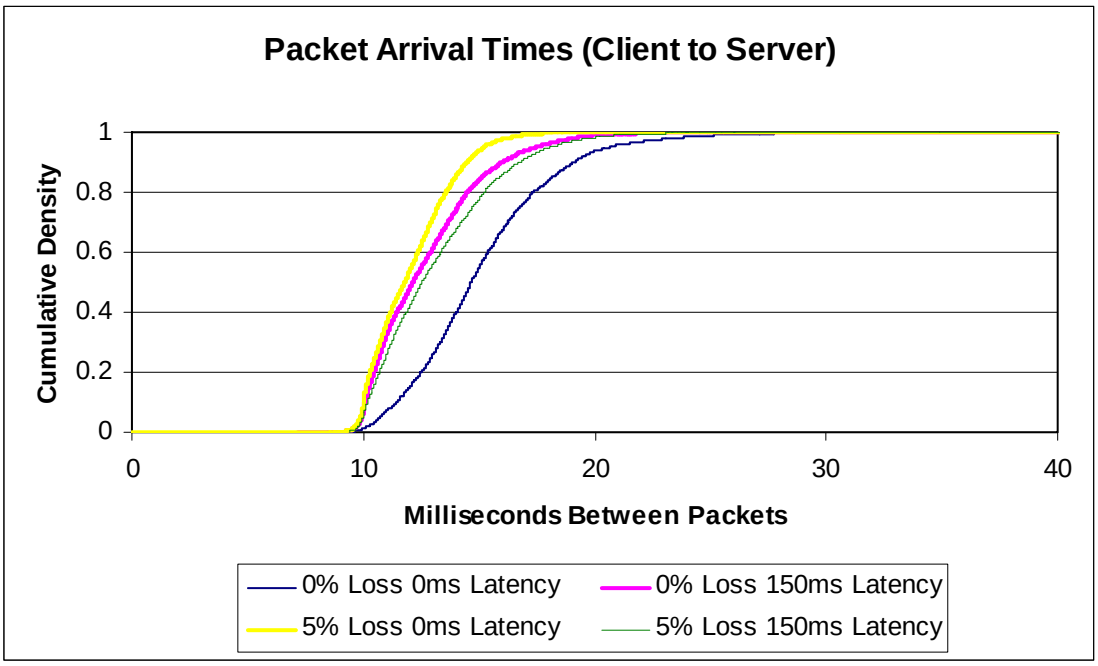

Figure 13: Packet Arrival Times (Client to Server)

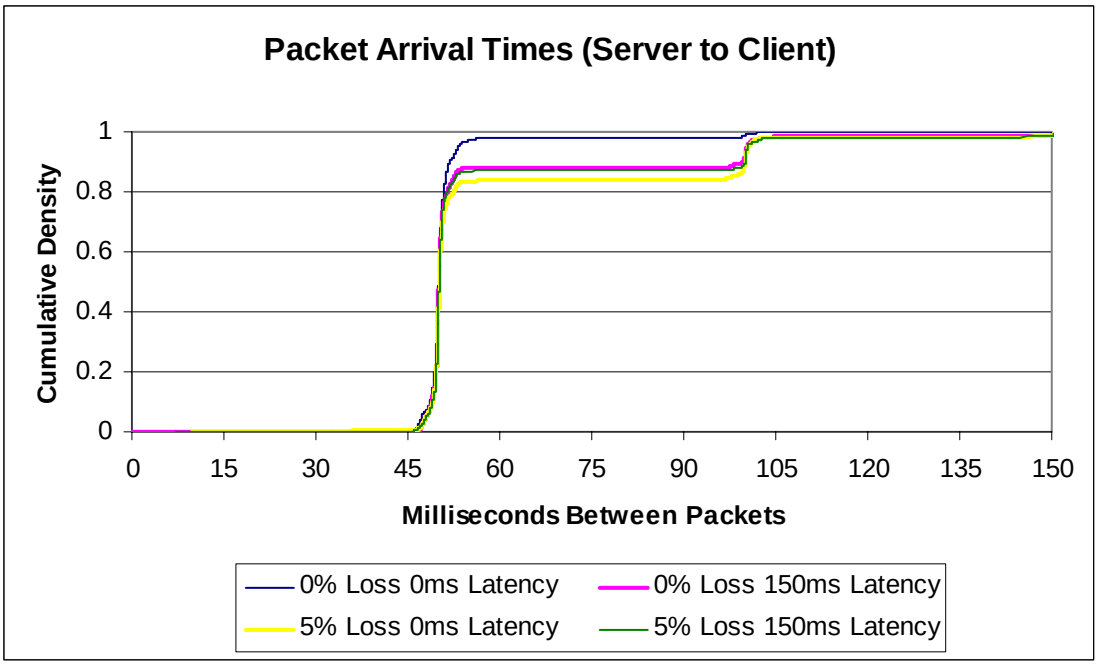

Figure 14: Packet Arrival Times (Server to Client)

Figure 13 shows the amount of time measured between packets that were sent by the client to the server and Figure 14 shows the amount of time measured between packets that were sent from the server to the client. This data was captured on the NIST Net router sitting between the client and the server. The graphs show that the client is somewhat inconsistent and sends packets every 10 to 20ms while the server is highly consistent and sends packets every 50ms (a small number have an interval of 100ms which is exactly twice the time of the regular interval).

7. Movement Tests

A player's ability to move his or her character around a game map is one of the most critical aspects of a FPS. The following tests were conducted in order to gauge the impact of network degradation, through both packet loss and latency, on a player's ability to move his or her character around a game map. Another goal of these tests was to determine how the game handles delayed or dropped packets in relation to a player's movements. Many FPS game servers use the player's last known position and velocity vector in order to keep track of character positions. This allows the server to tolerate more latency or packet loss at the expense of data accuracy [SKH01]. The game's behavior during the movement tests would indicate whether UT2003 employed a dead reckoning system to determine player positions.

7.1 Simple Movement

The goal of these tests was to determine what effects, if any, packet loss and latency had on a player's ability to perform the most basic game function of moving in a straight line, which we call the footrace test. Our test setup consisted of three client computers connected to the dedicated game server through the NIST Net router. Two of the clients raced in a straight line while the third client stood off to the side to observe which character crossed the finish line first. The test was conducted in the Tokara Forest map, which is included in a standard game install.

7.1.1 Packet Loss

The footrace test was conducted with one of the clients having symmetrical packet loss levels of .5%, 1%, 1.5%, 2.5%, and 10%. The clients raced three times at each level of loss. For all of these tests both the player with packet loss and the player without packet loss crossed the finish line at the same time.

7.1.2 Latency

The footrace test was also conducted with latency times of 50ms, 100ms, 150ms, 200ms and 300ms. As with the packet loss tests, we conducted the race three times for each amount of latency. These tests yielded results identical to those gathered from the packet loss tests, with both players crossing the finish line at the same time.

7.1.3 Summary

The results of these test show that packet loss and latency do not have any effect on a player's ability to perform simple movements in the game. Because of the fact that the results of the race were the same every time, we can infer that the majority of calculations for a player's location within the map are done on the client computer. Most likely the client simply sends a vector to the server specifying direction and speed. The server probably then assumes the player will continue in that direction at that speed until it receives different information, which means the game employs dead reckoning.

7.2 Complex Movement

Since the majority of movement in faced paced FPS games is not unidirectional we decided to conduct tests of more complicated movement. For these tests we defined a specific course through the Flux2 map that is included in a standard game install. The course was defined as follows:

1) Starting at the barrel by the armor (Figure 15), run along the catwalk, jump the corner to the left towards the door

Figure 15: Start of Complex Movement Test

2) Enter the door, and pick up the Shock Rifle, jump the corner to the left, jump the next left corner (Figure 16), pick up the health pack

Figure 16: Middle of Obstacle Course

3) Spin around and then jump down, pick up the chain gun and ammo, and walk out towards the door
4) Go straight and jump on the barrel, then translocate up to the platform with the double damage, pick up the double damage
5) Translocate to the link gun (Figure 17), walk down the ramp, quad jump up to steaming structure, jump off towards alcove, run up and over the alcove

Figure 17: Obstacle Course Continued

6) Translocate to top of crane, translocate to steam vent (Figure 18), jump off

7) The course ends when player hits the ground

Figure 18: End of Complex Movement Test

The player running the course was given various amounts of latency and packet loss and was timed with a stopwatch from start to finish. Based on our familiarity with the game genre, we felt that this selection of tasks provided an appropriate sampling of typical movements found in a regular game.

7.2.1 Packet Loss

To test the impact of packet loss we ran the obstacle course with symmetrical packet loss levels of 0%, 1%, 4%, and 6%. The average time to complete three tests at each of these packet loss levels were 53.67 seconds, 56.33 seconds, 51.00 seconds, 55.67 seconds respectively. Figure 19 illustrates the average of the three test times for each level of loss, shown with a 95% confidence interval. The figure clearly shows that packet loss had no noticeable effects on the course completion times.

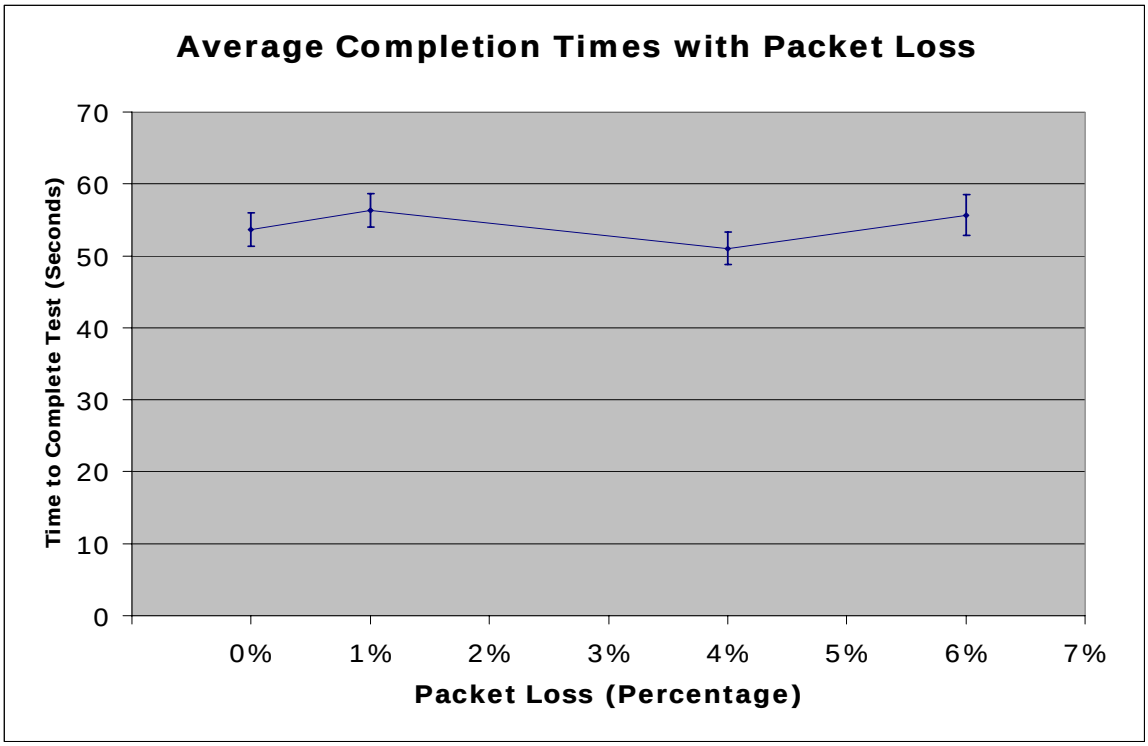

Figure 19: Complex Movement with Packet Loss

7.2.2 Latency

To test the impact of latency we ran the obstacle course with latency amounts of 0ms, 100ms, 150ms, 200ms, 300ms, 250ms, 400ms. The averages of the three completion times for each latency level are 52.33 seconds, 50.33 seconds, 50.33 seconds, 50.00 seconds, 52.00 seconds, 52.33 seconds, and 57.00 seconds respectively. Figure 20 illustrates the average of the three test times for each level of latency, shown with a 95% confidence interval. The figure shows that latency had no noticeable effects on the course completion times up to latency amounts of 300ms where a slight upward trend can be seen that continues through the tests at 400ms.

The results of these tests indicated that latency has very little impact on complex maneuvers. Although a slight upswing in times can be seen at the extremely high latency levels, latencies of that magnitude are rarely encountered in real world situations.

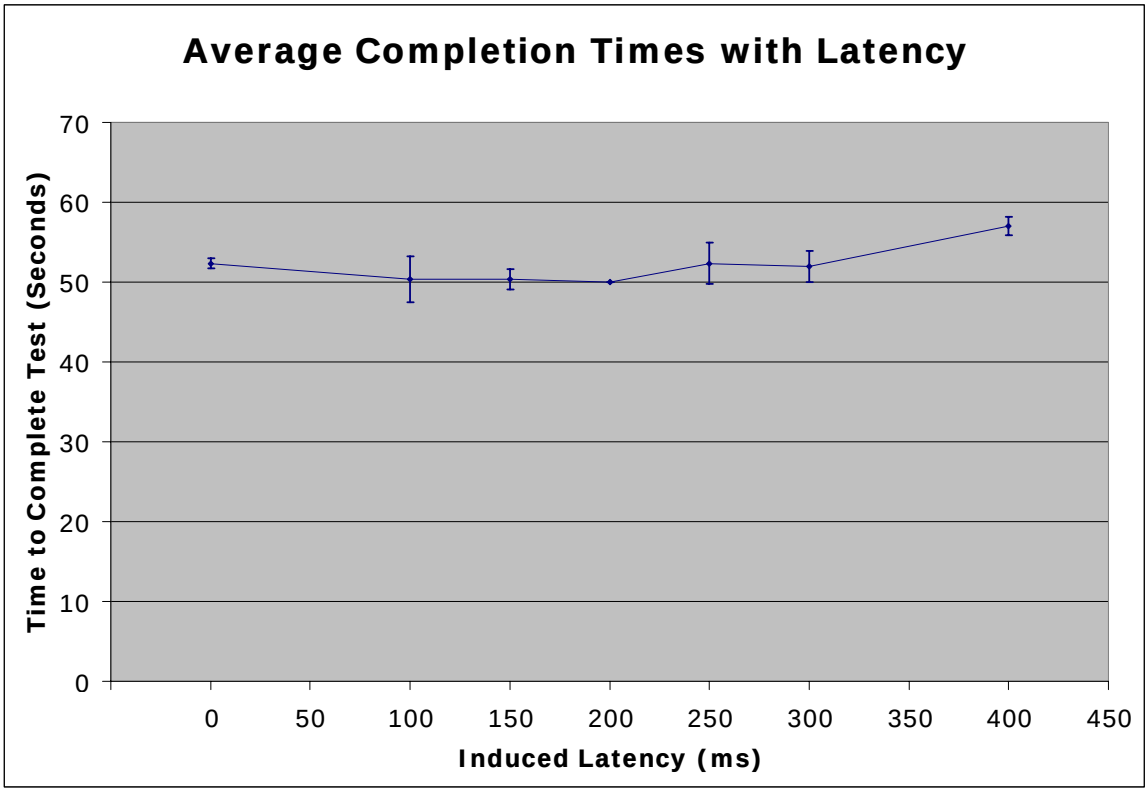

Figure 20: Complex Movement with Latency

7.2.3 Summary

The results of these tests indicate that neither latency nor loss have a noticeable impact on a player's ability to perform complex movements in the game environment. These results, combined with the results of our simple movement tests, lead us to believe that almost all computations relating to player movement are done by the client's computer. This leaves the server available to keep track of player shots, record player statistics, and transmit player locations to other players, and hence improve scalability.

8. Precise Shooting Tests

Complimentary to movement in any FPS game is shooting. For the majority of FPS games, shooting can be broken down into two subsections: normal shooting and precision shooting. Precision shooting is much less forgiving when it comes to network degradation than is normal shooting. In this section we have isolated precise shooting in order to learn the effects of both packet loss and latency, and how they affect both user performance and satisfaction.

To conduct our experiments we ran a series of tests using the map CTF-Face3 and limited weapon use to that of only the Lightning Gun. Player 1 obtained the Lightning Gun and positioned himself on top of one of the 2 identical towers, facing the center pyramid. Player 2 would then move into position at the base of the center pyramid. When the experiment began, Player 1 would have 10 minutes to hit Player 2 as many times as possible with his fully zoomed Lightning gun, while Player 2 tried to avoid Player 1 by means of running, side-stepping and jumping. If Player 2 was killed he would immediately return to the pyramid and begin dodging Player 1. Similarly if Player 1 ran out of ammunition he would quickly obtain more in the adjacent room.

8.1 Packet Loss

The first round of experiments we performed was used to gauge the effects of packet loss on precise shooting. We began by conducting a baseline experiment where the player doing the shooting, Player 1, had no loss at all. After gathering the number of hits and misses, we then increased the packet loss percentage to 1% and 3%. We then repeated the full experiment 3 times for each of 2 players, and averaged the results, seen

in Figure 21. For each experiment we used a standard latency of 100ms to simulate the conditions of an average online gaming experience.

At the baseline of 0% packet loss the average hit fraction was .314 for all experiments. After increasing the packet loss to rates of 1% and 3% the accuracy's changed to .334 and .319 respectively. In order to better analyze the results, confidence intervals around each data point were calculated. After examining Figure 21 it is clear that the confidence intervals for 0%, 1% and 3% packet loss all overlap which implies that there is no statistical difference between these points.

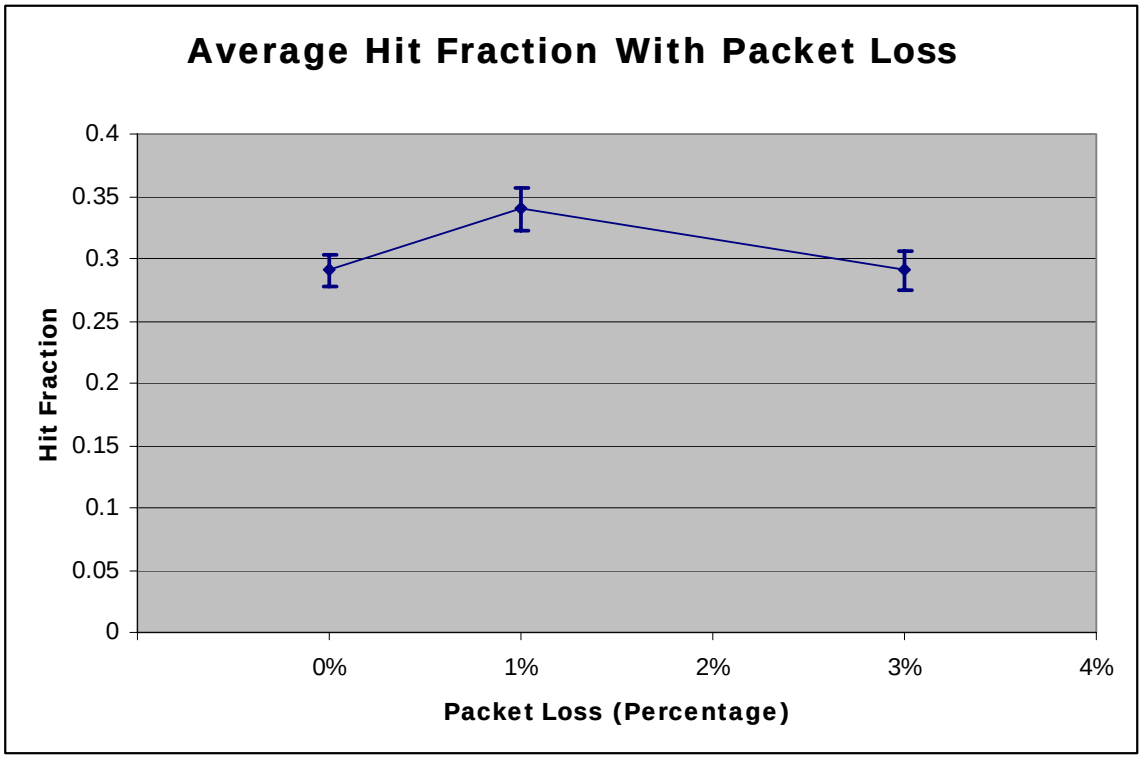

Figure 21: Precise Shooting with Packet Loss

8.2 Latency

To further examine the effects of network conditions on precise shooting we conducted our experiments again. However this time, we kept packet loss at a constant 0%, while we increased the latency on the person firing. Again, we ran this test using two different players, and a total of three times for each player, per latency level.

The average hit percentages with 0ms, 50ms and 75ms are approximately .45, .43 and .43 respectively, seen in Figure 22. A first order linear regression is also shown in the figure, clearly illustrating the downward trend as latency increased. As the case with packet loss, these data points are not statistically different. However after 100ms there is a sharp change in the data. At 100ms latency, the average accuracy drops to approximately .33, down almost 10% from previous experiments. As latency increases, shot accuracy continues to decline, reaching .26 and .19 at 200ms and 300ms. Additionally, it can be observed that while the confidence intervals of the 0ms, 50ms and 75ms experiments overlap and are thus statistically indistinguishable; the confidence intervals of the 100ms, 200ms and 300ms experiments are mutually exclusive and hence statistically different.

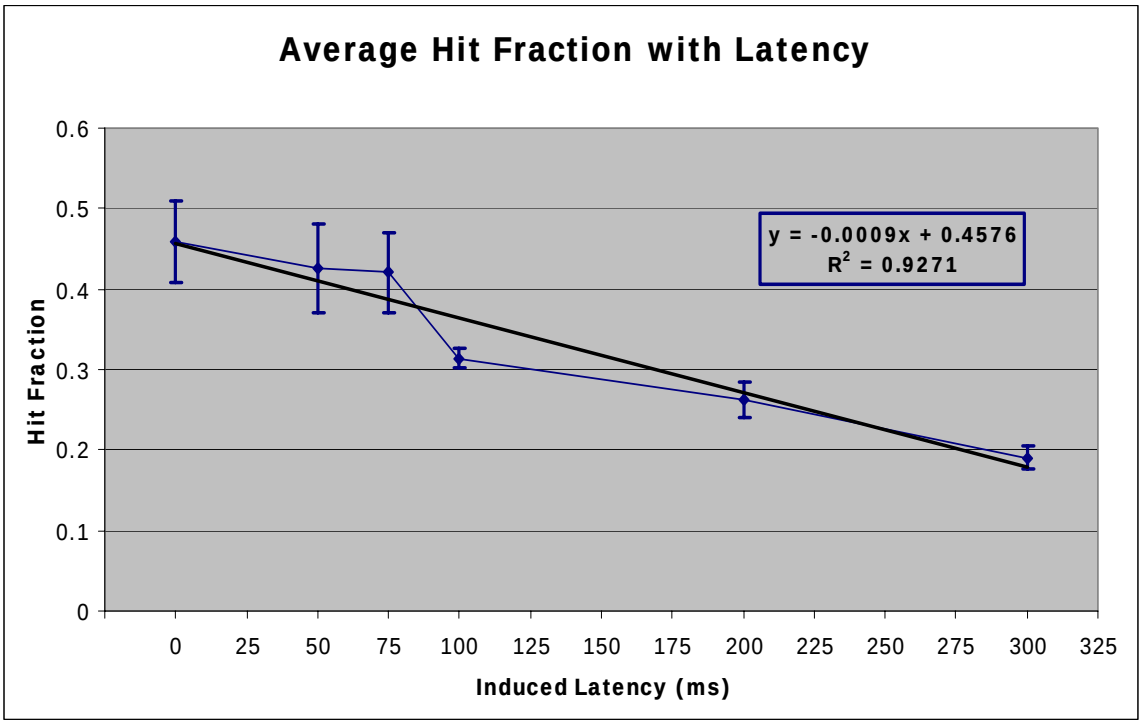

Figure 22: Precise Shooting with Latency

8.3 Summary

Judging from the results of these tests, we can conclude that precision shooting is robust enough to not be effected by any normal rate of packet loss found in an online FPS game. We assume this is because the number of packets containing shot data represents such a small percentage of all transmitted packets that it will not affect a player's ability to aim with precision. However, we found that the amount of latency can significantly affect a user's ability to perform precision shooting. Overall, going from an ideal situation with 0ms latency, to a less desirable one with 300ms, the average shot accuracy drops by over 50%.

9. Restricted Deathmatch Test

After completing the experiments testing shot accuracy and movement independently of each other, we conducted tests that combined the two. We set up a Deathmatch game between one human player and one computer-controlled bot. The intsta-gib mutator was used on the map. This forced the user to have to aim and dodge because they could not send bullets in a wide spread or gain life to avoid damage from the bot.

We conducted the experiments using the Training Day map because it is small enough for two players to find each other easily, but provided enough cover so as to prevent an individual from gaining too much of an advantage by “spawn camping” (killing the other player as soon as they come back to life). The same bot, “Widowmaker,” was used for every test to avoid discrepancies resulting from the different fighting styles associated with the different characters. After each 5-minute match, we recorded the number of kills and deaths accumulated by the human player.

9.1 Packet Loss

We administered the experiment with packet loss levels of 0%, 1%, 2%, and 5%. In this test the change in packet loss made no noticeable difference in gameplay. Each packet loss level was tested four times, once for each user. Figure 23 shows a graph of the average number of kills and deaths at each packet loss level along with error bars for a 95% confidence interval. All of the data points fall within the confidence intervals of each other. For this reason we can conclude that packet loss has no quantifiable performance effects on real world gameplay.

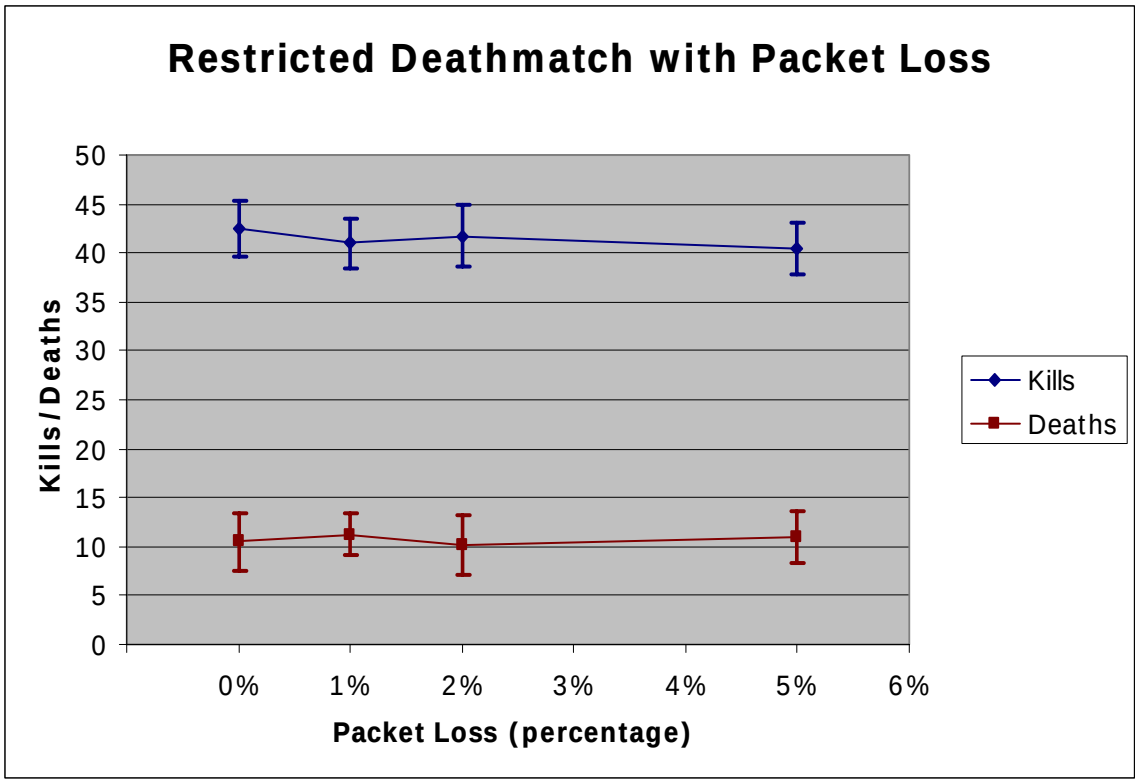

Figure 23: 5 Minute Restricted Deathmatch with Packet Loss

9.2 Latency

This test was run four separate times at each latency level. Looking at the average number of kills and deaths from Figure 24 for 0ms, 50ms, 75ms, and 100ms of latency there is a decrease in player performance. After looking at the 95% confidence interval it is obvious that the intervals overlap and for that reason we can say that there is no difference between the first four latency levels. Comparing the 0ms latency result to the 150ms result shows a distinct change. The number of kills is significantly lower and deaths increase, but more importantly the confidence intervals do not overlap and for this reason we can say that these two latency levels affect performance. The figure also shows first order linear regressions plotted for both kills and deaths. These lines clearly

show the sharp downward trend for number of kills and slight upward trend for number of deaths.

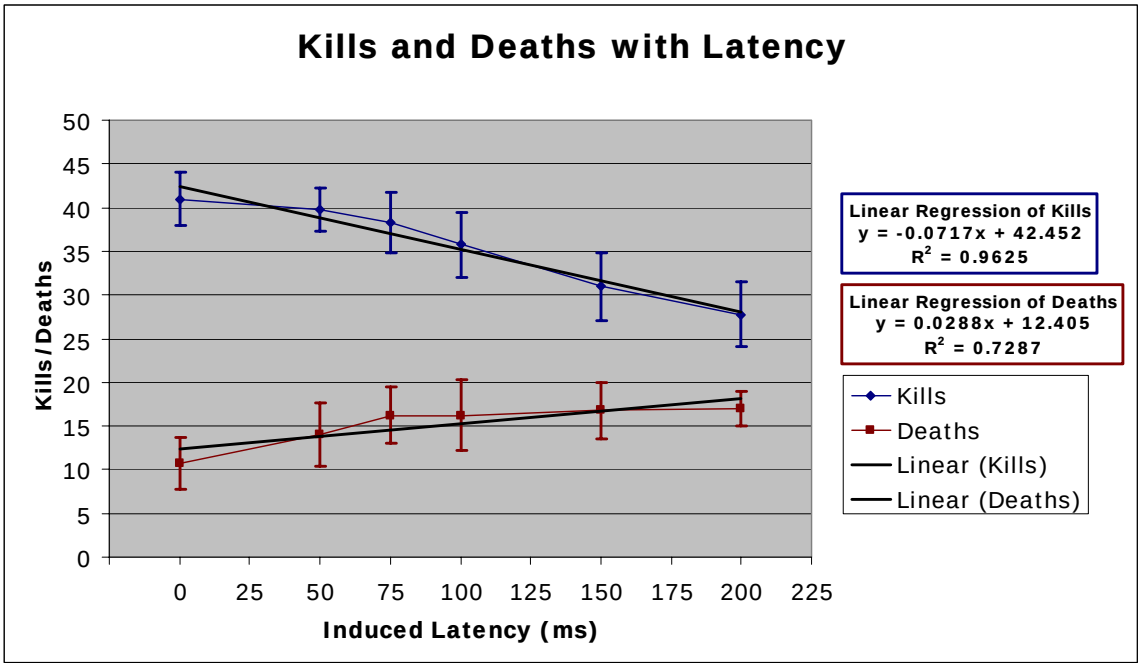

Figure 24: 5 Minute Restricted Deathmatch with Latency

9.3 Summary

As with previous tests we found that packet loss has no noticeable impact on player performance. With packet loss a player's reflexes are not impaired. Due to the fast pace of the game, the player is unlikely to notice a small percentage of dropped packets, and the player's overall skill is not affected. However, latencies above 100ms caused both number of kills to drop and number of deaths to increase. This indicates that the delay in feedback for the combination of shooting and movement has a detrimental impact on the player's score.

10. Full Deathmatch Test

After completing all the previous tests on individual aspects of the game we wanted to determine what impact packet loss and latency had on actual player scores in regular games. To do this we set up a dedicated server running the same Training Day map as was used in the previous tests, only this time there were no weapon restrictions so players were free to use whichever weapon they thought would allow them to do their best.

As was the case before, the game was limited to two players, one human and one computer controlled bot. As in the previous Deathmatch tests, the bot named "Widowmaker" was used for each game. After each 5-minute match we recorded the number of kills and deaths accumulated by the human player.

10.1 Packet Loss

We administered the experiment with packet loss levels of 0%, 1%, 2%, and 5%. For each level of packet loss, the test was conducted four separate times. Figure 25 shows a graph of the average number of kills and deaths at each packet loss level along with error bars for a 95% confidence interval. Since all confidence intervals overlap, we can say that packet loss has no statically measurable impact on real world gameplay.

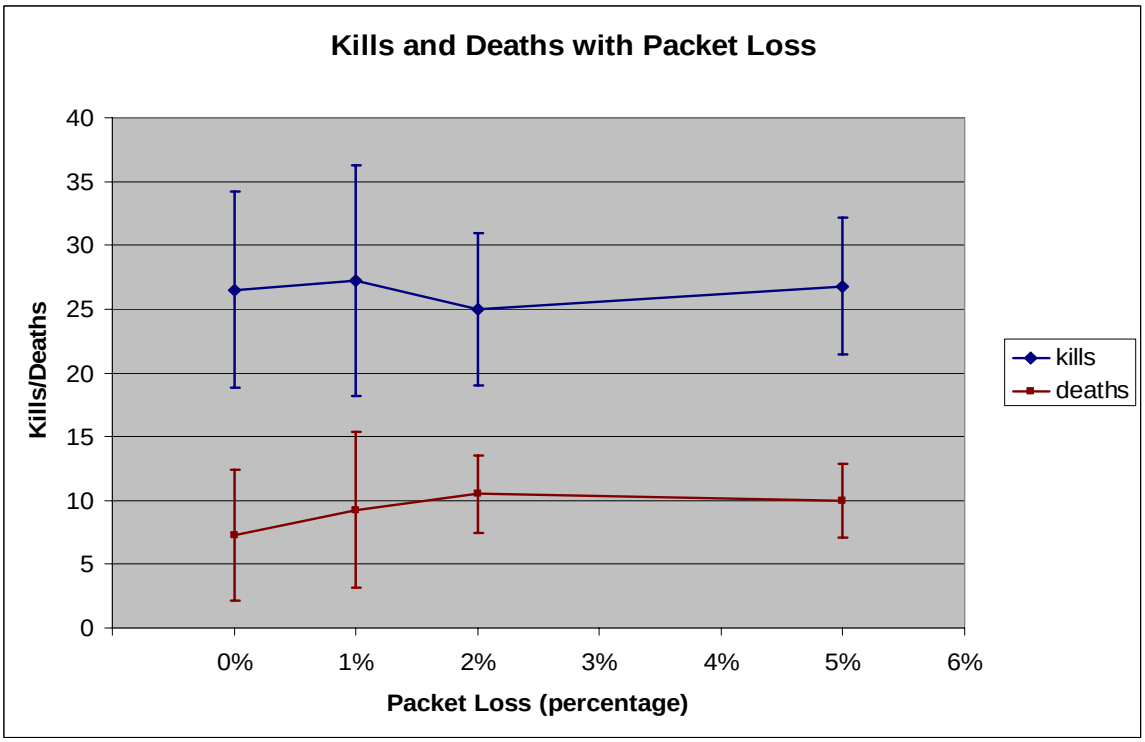

Figure 25: 5 Minute Deathmatch with Packet Loss

10.2 Latency

We also administered the test with latency amounts of 0ms, 50ms, 75ms, 100ms, 150ms, and 200ms. The test was run four separate times at each latency level. The 95% confidence interval in Figure 26 shows that latency has limited statistical impact on the outcome since most data points fall within the error ranges of the other data points. Slight sloping trends can be seen in the graph, with the number of deaths rising and kills dropping; however, since the confidence intervals range over the majority of the points these trends can not be counted as having a definitive impact on player scores. However, the figure also shows the first order linear regressions for both kills and deaths. These lines show that, despite the overlapping confidence intervals, there are definite trends in the data that should not be overlooked.

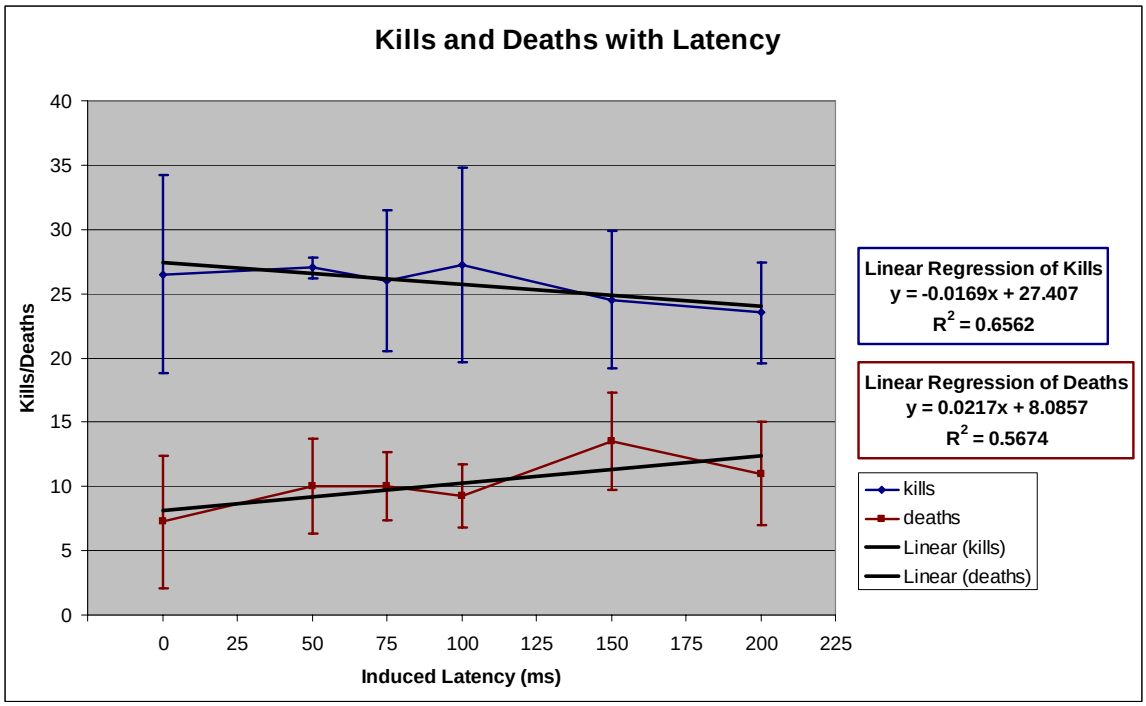

Figure 26: 5 Minute Deathmatch with Latency

10.3 Summary

In these tests we found that neither packet loss nor latency affected player performance. Even though players were able to notice the game lagging with the higher latency levels, the results did not indicated a negative impact on player scores. These results may be attributed to the fact that players can compensate for high latencies by switching to weapons that require less accurate aiming. Certain weapons such as the Flak Cannon fire in a cone shape that spreads out as it travels away from the shooter. With a high amount of latency the player only needs to aim in the general direction of his or her opponent to cause a substantial amount of damage.

11. Subjective Quality Assessment

Over the course of our tests we noticed degradation in the player's subjective opinion on the quality of the gameplay. As the tests progressed we came to realize that players were able to notice when latencies as low as 75ms were induced on their connection, and found gameplay to became less enjoyable at latencies over 100ms. Even though the player scores were not statistically impacted, players complained of game lag and felt as if they were performing worse.

Occasionally players were also able to notice packet loss when testing with a loss rate of 3%. The game would sometimes not display animations for shots fired, but this had no affect on player performance. Most of the time, however, players were completely unaware of any induced packet loss.

Players were unable to notice any latency or packet loss in the simple movement tests (running in a straight line) and were only marginally aware of a slight delay in the complex movement tests. The significant subjective impact was noticed during the precise shooting tests. Players were extremely aggravated when trying to aim and shooting when latency amounts higher than 100ms were induced on their connections. Also, during the restricted and unrestricted full game tests, players found high levels of latency to be annoying because the game would not react as quickly as the players wanted it to. This was particularly prevalent in the full game tests. Players felt as if they were performing worse, even if though their scores did not reflect it.

Generally it would be wise for players to avoid servers with ping times over 150ms and packet loss levels over 3%. Even though they do not significantly impact the

player score, they do make gameplay less enjoyable, which partially defeats the purpose of playing the game in the first place.

12. Conclusions

First Person Shooter (FPS) games are very sensitive to changes in network performance. Latency changes and lost packets could mean that your target is no longer where you expected him to be or you bullet could never even fire. For these reasons it is very important to understand what network problems cause the user the most harm. To find this out we tested UT2003, a popular FPS game, using a variety of tests that dealt with movement, shooting, or a combination of the two. We ran each test for a variety of latencies and packet loss amounts to see what affected performance and how much.

There were two different movement tests that we examined. The first dealt with a simple race between two players, one with latency or packet loss and one without. For both sets of these tests there was no difference in when each player crossed the finish line, therefore we conclude that neither latency nor packet loss has any affect on straight movement. In the second movement test a player ran an obstacle course multiple times with varying latency or packet loss. In these tests when the player had packet loss his performance, while varied, remained within the confidence interval. When latency was applied to the player his times for the obstacle course increased. Additionally the confidence intervals for the different latency settings did not overlap at high latencies (300ms and above), which shows that there is a difference in performance across those latency settings. For lower latency settings the confidence intervals did overlap, which shows no statistical difference.

For pure shooting we conducted a test where one player used a sniper rifle from a tower and tried to hit a moving player as many times as he could in 10 minutes. When packet loss was added the sniper's amount of hits did not decrease a noticeable

percentage. When latency was added, shot accuracy dropped from 50%, with 0ms, to less than 20% with 200ms.

Finally we performed two full game tests. In the first test one player fought against one bot and tried to kill him with a precision weapon. In this experiment, across the different packet loss percentages the amount of times the player killed the bot and the amount of times the player got killed by the bot stayed within the confidence interval. When higher levels of latency were given to the player his number of kills started to go down and his deaths went up. There is a distinct difference in performance between 0ms latency and 150ms latency. The second full game test used the same map and bot as the first one but allowed the player to use any of the weapons on the map, which included some less precise high damage weapons. In these tests the amount of kills and deaths for the player remained within the confidence intervals across both tests. This could be mainly due to the fact that with less precise weapons the player does not have to be as accurate with his shots and can still perform well with higher latency or packet loss.

At the end of these tests a few things became clear. The first one is that packet loss does not affect FPS games significantly. While changes in packet loss did prevent some bullets from reaching their intended targets, FPS games often expect players to shoot off numerous bullets with the assumption that many of them will not hit their targets. For this reason losing an occasional bullet did not affect the game significantly.

Latency values proved to affect UT2003 far more than packet loss. Higher latency means that all of the actions that a player performs are slowed. Players performing delicate tasks such as hitting a target with a sniper rifle have to aim ahead of the target to compensate for the slower response time from their weapons. Because of

this we conclude that latency has a much stronger impact on user performance than packet loss.

13. Future Work

Over the course of working on our project we came across several topics that we felt merited further research. One of which was the amount of variance in latency levels. We believe that constantly changing latencies would make it more difficult to perform actions in the game, in particular precise aiming and complicated movements. With constant latency at the low to moderate amounts we saw that users were usually able to compensate for the delay without having a significant impact on performance. If these were no longer constant and fluctuating rapidly we believe adaptation would be tougher and make game-play much more difficult.

Also, when packets on a network are lost, they usually are not dropped completely at random. Instead, they are usually lost in bursts, with several packets in a row being lost. We think that the concept of "bursty loss" merits further research. Perhaps the game software can handle randomly lost packets but we would like to see what happens when groups of packets are lost at a time.

Another topic we thought would be interesting to investigate was how the results of our tests compare to the same tests run on a different FPS game from a different lineage. By comparing our results, to results obtained by similar experiments we could determine how changes at the network level affect game performance and user satisfaction.

14. References

[A01] Grenville Armitage. When Does the Quake III Community Play? May, 2001. [Online] http://gja.space4me.com/things/quake3-when-051401.html

[A02] Grenville Armitage. Lag over 150 milliseconds is unacceptable. Internet Measurement Workshop, 2002. [Online] http://gja.space4me.com/q3/imw2001/poster110101.pdf

[FCF+03] Wu-Chang Feng, Francis Chang, Wu-chi Feng, Jonathan Walpole. Provisioning On-line Games: A Traffic Analysis of Popular First-Person Shooters.

[FCF+02] Wu-chang Feng, Francis Chang, Wu-chi Feng, Jonathan Walpole. Provisioning On-Line Games: A Traffic Analysis of a Busy Counter-Strike Server. Proceedings of the Internet Measurement Workshop, 2002. [Online] http://www.cse.ogi.edu/sysl/projects/cstrike/CSE-02-005.pdf

[H01] Tristan Henderson. Latency and user Behaviour on a Multiplayer Games Server. Proceedings of NGC 2001, London, UK, pp1-13, November 2001. [Online] http://www.cs.ucl.ac.uk/staff/t.henderson/docs.html

[H03] Tristan Henderson. The Effects of Relative Delay in Networked Games. University of London, April 2003. [Online] http://www.cs.ucl.ac.uk/staff/t.henderson/docs.html

[LW01] Dave LaPointe and Josh Winslow. Analyzing and Simulating Network Game Traffic. Worcester Polytechnic Institute, 2001. [Online] http://www.cs.wpi.edu/~claypool/mqp/net-game/game.pdf

[PW02] Lothar Pantel and Lars C. Wolf. On the Impact of Delay on Real-Time Multiplayer Games. Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video, Miami, Florida, 2002. [Online] http://portal.acm.org/citation.cfm?doid=507670.507674

[SEB03] Nathan Sheldon, Eric Girard, Seth Borg. The Effect of Latency on User Performance in Warcraft III. Worcester Polytechnic Institute, 2003. [Online] http://www.cs.wpi.edu/~claypool/mqp/war3/mqp.pdf

[SKH01] Jouni Smed, Timo Kaukoranta, Harri Hakonen. Aspects of Networking in Multiplayer Computer Games. University of Turku, 2001. [Online] http://staff.cs.utu.fi/~jounsmed/papers/AspectsOfMCGs.pdf

A. Data Tables

A.1 Complex Movement Test Data

A.1.1 Complex Movement with Packet Loss

Packet Loss	Trial 1	Trial 2	Trial 3	Average	std dev	confidence
0	56	53	52	53.6666667	2.081666	2.35558355
0.01	58	57	54	56.3333333	2.081666	2.35558355
0.04	53	51	49	51	2	2.26317147
0.06	56	58	53	55.6666667	2.51661148	2.84776165

A.1.2 Complex Movement with Latency

Total Latency (ms)	Trial 1	Trial 2	Trial 3	Average	std dev	confidence
0	52	53	52	52.333	0.57735027	0.653321
100	48	53	50	50.333	2.51661148	2.84776
150	51	51	49	50.333	1.15470054	1.306642
200	50	50	50	50	0	0
250	51	55	51	52.333	2.30940108	2.613284
300	51	51	54	52	1.73205081	1.959963
400	58	57	56	57	1	1.131585

A.2 Precise Shooting Data

A.2.1 Precise Shooting with Packet Loss

loss	hits	misses	total shots	hit percentage	standard dev	confidence interval
0	18	36	54	0.333333333		
0	16	33	49	0.326530612		
0	19	41	60	0.316666667		
0	19	43	62	0.306451613		
0	20	45	65	0.307692308		
0	18	44	62	0.290322581	0.01545178	0.012363772
1	18	31	49	0.367346939		
1	22	50	72	0.305555556		
1	20	42	62	0.322580645		
1	19	36	55	0.345454545		
1	20	42	62	0.322580645		
1	17	33	50	0.34	0.021675221	0.01734347
3	20	44	64	0.3125		
3	18	40	58	0.310344828		
3	19	39	58	0.327586207		
3	17	32	49	0.346938776		
3	20	42	62	0.322580645		
3	18	44	62	0.290322581	0.018996634	0.015200194

A.2.2 Precise Shooting with Latency

latency (ms)	hits	misses	total shots	hit percentage	standard dev	confidence interval
0	30	29	59	0.508474576		
0	28	27	55	0.509090909		
0	27	31	58	0.465517241		
0	31	27	58	0.534482759		
0	31	30	61	0.508196721		
0	26	27	53	0.490566038	0.063591144	0.050882577
50	14	26	40	0.35		
50	15	25	40	0.375		
50	15	25	40	0.375		
50	18	22	40	0.45		
50	19	21	40	0.475		
50	21	19	40	0.525	0.068920244	0.055146667
75	18	22	40	0.45		
75	17	23	40	0.425		
75	12	28	40	0.3		
75	18	22	40	0.45		
75	19	21	40	0.475		
75	17	23	40	0.425	0.062081935	0.049674981
100	18	36	54	0.333333333		
100	16	33	49	0.326530612		
100	19	41	60	0.316666667		
100	19	43	62	0.306451613		
100	20	45	65	0.307692308		
100	18	44	62	0.290322581	0.01545178	0.012363772
200	16	43	59	0.271186441		
200	15	47	62	0.241935484		
200	17	40	57	0.298245614		
200	15	45	60	0.25		
200	14	47	61	0.229508197		
200	17	42	59	0.288135593	0.027112039	0.021693751
300	10	48	58	0.172413793		
300	12	47	59	0.203389831		
300	13	49	62	0.209677419		
300	11	48	59	0.186440678		
300	12	47	59	0.203389831		
300	10	49	59	0.169491525	0.017230624	0.013787117

A.3 Restricted Deathmatch Data

A.3.1 Packet Loss

Loss	Kills	Deaths	Average Kills	Average Deaths	Confidence Kills	Confidence Deaths
0	43	9				
0	41	11				
0	39	7				
0	47	15	42.5	10.5	2.898824056	2.898824056
1	45	12				
1	41	14				
1	38	11				
1	40	8	41	11.25	2.498472125	2.121721955
2	42	12				
2	37	5				
2	42	11				
2	46	13	41.75	10.25	3.12789122	3.05016749
5	36	15				
5	43	12				
5	42	8				
5	41	9	40.5	11	2.638680657	2.683789576

A.3.2 Latency

Latency	Kills	Deaths	Average Kills	Average Deaths	Confidence Kills	Confidence Deaths
0	43	7				
0	36	15				
0	44	9				
0	41	12	41	10.75	3.020505513	2.970410738
50	39	11				
50	36	19				
50	42	10				
50	42	16	39.75	14	2.437672938	3.600681558
75	39	14				
75	33	20				
75	43	12				
75	38	19	38.25	16.25	3.490646403	3.277814365
100	38	11				
100	30	18				
100	40	14				
100	35	22	35.75	16.25	3.691227114	4.062788231
150	34	14				
150	25	20				
150	35	13				
150	30	20	31	16.75	3.858190605	3.203729896
200	30	15				
200	22	19				
200	32	15				
200	27	19	27.75	17	3.691227114	1.959962787

A.4 Full Game Data

A.4.1 Full Game with Packet Loss

Kills	Deaths	Loss	Average Kills	Average Deaths	std dev kills	std dev deaths	confidence kills	confidence deaths
28	7	0						
21	10	0						
20	12	0						
37	0	0	26.5	7.25	7.85281266	5.251983752	7.695610295	5.146846357
28	11	1						
21	13	1						
20	13	1						
40	0	1	27.25	9.25	9.215023965	6.238322424	9.030557544	6.113443638
26	12	2						
22	11	2						
19	13	2						
33	6	2	25	10.5	6.055300708	3.109126351	5.934085652	3.046887836
25	10	5						
22	13	5						
21	11	5						
33	6	5	26.8	10	5.439056291	2.943920289	5.33017722	2.88498887

A.4.2 Full Game with Latency

Kills	Deaths	Latency	Average Kills	Average Deaths	confidence kills	confidence deaths
28	7	0				
21	10	0				
20	12	0				
37	0	0	26.5	7.25	7.695614995	5.146849501
26	10	50				
28	9	50				
27	15	50				
27	6	50	27	10	0.800151946	3.66675686
28	8	75				
22	9	75				
21	14	75				
33	9	75	26	10	5.48556537	2.65380378
27	9	100				
24	8	100				
20	11	100				
38	5	100	27.25	9.25	7.564504633	2.449954981
23	14	150				
18	15	150				
26	17	150				
31	8	150	24.5	13.5	5.337679232	3.795453936
26	10	200				
23	9	200				
18	17	200				
27	8	200	23.5	11	3.960550069	4.00075973