The Effects of Loss and Latency on User Performance in Unreal Tournament 2003®

Tom Beigbeder, Rory Coughlan, Corey Lusher, John Plunkett,
Emmanuel Agu, Mark Claypool
Computer Science Department at Worcester Polytechnic Institute
100 Institute Road, Worcester, MA 01609, USA
{emmanuel|claypool}@cs.wpi.edu

ABSTRACT

The growth in the popularity of interactive network games has increased the importance of a better understanding of the effects of packet loss and latency on user performance. While previous work on network games has studied user tolerance for high latencies and has studied the effects of latency on user performance in real-time strategy games, to the best of our knowledge, there has been no systematic study of the effects of loss and latency on user performance. In this paper we study user performance for Unreal Tournament 2003 (UT2003), a popular first person shooter game, under varying amounts of packet loss and latency. First, we deduced typical real world values of packet loss and latency experienced on the Internet by monitoring numerous operational UT2003 game servers. We then used these deduced values of loss and latency in a controlled networked environment that emulated various conditions of loss and latency, allowing us to monitor UT2003 at the network, application and user levels. We designed maps that isolated the fundamental first person shooter interaction components of movement and shooting, and conducted numerous user studies under controlled network conditions. We find that typical ranges of packet loss have no impact on user performance or on the quality of game play. The levels of latency typical for most UT2003 Internet servers, while sometimes unpleasant, do not significantly affect the outcome of the game. Since most first person shooter games typically consist of generic player actions similar to those that we tested, we believe that these results have broader implications.

Categories and Subject Descriptors: C.2 Computer Communication Networks: Distributed Systems

General Terms and Keywords: Performance, Experimentation, Network Games, Loss, Latency

1. INTRODUCTION

In recent years, the declining costs of increasingly powerful personal computers has increased their acquisition rates and created a growing user base for computer games. The increase in residential broadband Internet connections with high capacities and low latencies have encouraged more and more game developers to incorporate multi-player features into their products. An understanding of how network related issues, such as latency and packet loss, affect the usability of games can be of great use to the companies that make these games, network software and equipment manufacturers, Internet Service Providers (ISPs), and the research community at large. In particular, once established latency requirements and any associated trade-offs are known, ISPs can establish tariffs based on customers' indicated maximum delays, requested Quality of Service (QoS), and the ISP's ability to meet these demands. Moreover, experimental study of network games can provide the data required for accurate simulations, a typical tool for evaluating network research, as well as insight for network architectures and designs that more effectively accommodate network game traffic turbulence.

While there has been research qualitatively characterizing the effects of latency for car racing [11], custom games [12], and real-time strategy games [13] as well as a general awareness of latency issues [3, 4, 8, 10], work on the effects of latency in popular First Person Shooter (FPS) games [1, 6] has not quantified its impact on player performance. In concentrating on the effects of latency on FPS games, the possibility that packet loss may be the bottleneck in performance for some network conditions may be overlooked. To the best of our knowledge, there have been no systematic studies of packet loss on user performance in FPS games. The study of loss on network games is increasingly important as wireless channels, more prone to packet loss than traditional wire-line environment, become widely adopted.

In general, the most popular FPS games have descended from two game lineages, using either a Quake or Unreal-based game engine [5]. As previous research has concentrated on FPS games derived from Quake, we used Epic Game's award winning[1] Unreal Tournament 2003[2] (UT2003) in our experiments. UT2003 is very popular, with approximately 1700 servers and 4400 players online at any given time.[3]

First, we deduced typical real world values of packet loss

SIGCOMM'04 Workshops Aug. 30+Sept. 3, 2004, Portland, Oregon, USA.
Copyright 2004 ACM 1-58113-942-X/04/0008 ...$5.00.

[1] Winner of "Best of Show" at the Electronic Entertainment Expo, Los Angeles California, May 2002.

[2] http://www.unrealtournament.com/ut2003/

[3] Gamespy, October 2003

and latency experienced on the Internet by monitoring operational UT 2003 game servers. We then used these values as guidelines for induced loss and latency values in a controlled emulated environment we designed, allowing us to monitor UT2003 at the network, application and user levels. We divided user interaction in UT2003 into fundamental FPS interaction components in order to isolate particular facets of game-play. These interaction components include movement, precision shooting, general shooting, and moving and shooting simultaneously. We designed experiments with game maps that allowed us to isolate each game component. Using our testbed, we ran numerous user studies during which we systematically changed the loss and latency and measured the impact on performance.

We find that the levels of packet loss and latency typically encountered on the Internet, while sometimes unpleasant, do not drastically impact user performance in UT2003. Loss, in particular, goes unnoticed and does not measurably affect user interaction. Latencies as low as 100 ms, on the other hand, can significantly degrade performance for shooting with precision weapons both in terms of accuracy and game responsiveness. Although the effects of latency on user performance in full UT2003 games with all components is less noticeable, there is still a clear user performance degradation trend as latency increases. Moreover, UT2003 feels sluggish when latencies are 150 ms and higher.

The rest of this paper is organized as follows: Section 2 presents background information on UT2003; Section 3 describes our approach to measure the effects of latency and loss on user performance in UT2003; Section 4 analyzes the application, network and user results from our experiments; Section 5 summarizes our conclusions; and Section 6 presents possible future work.

2. BACKGROUND

First Person Shooters (FPS) are games in which a user interacts with the game world through the eyes of a virtual character (the "first person"), collects weapons and attempts to destroy other players (the "shooter"). Unreal Tournament 2003 (UT2003) is an online FPS in which up to 32 players can compete simultaneously on a single server over the Internet. Players navigate pre-defined obstacle courses or maps while playing the game. There are over 35 indoor and outdoor maps that come with UT2003, while many more user-created maps can be acquired from Web sites or simply by joining a server running a custom map.

There five multi-player modes UT2003 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. *Team Deathmatch* is very similar to Deathmatch except that instead of a complete free-for-all, the players are split into two teams and the team whose players have the highest combined score wins. *Capture the Flag* also pits two teams against each other, but the teams try to protect their own flag while trying to capture the opposing team's flag. The match ends when either one team achieves a pre-defined number of captures or a time limit expires. In *Double Domination*, teams fight to capture and control specific key points of a map with scores awarded for each of these points that a team captures. *Bombing Run* features a futuristic football style match with teams passing and running to either cross a goal for 7 points, or shoot the ball into the goal for 3 points. Since the most popular modes of play are Deathmatch and Capture the Flag, we used Deathmatch and Capture the Flag maps for all our tests. However, despite the slight differences in game play modes we believe our results would pertain to other modes as well.

Like many FPS games, UT2003 includes a large assortment of weapons. Examples include the Mini-gun which is capable of firing high volumes of bullets in a very short time, the Flak Cannon, which can scatter shards of metal in the general vicinity of opponents, and the Rocket Launcher, which can load and launch up to three rockets at a time. While all these weapons destroy opponents, we hypothesize they differ in their timing requirements based on the precision required in aiming. Table 1 shows a table of weapons broken into four categories based on our subjective view of the amount of precision that is required to use them. We hypothesize that higher precision weapons tend to be more difficult to use effectively when network packets are either lost or delayed, while weapons that require less precision are less affected by lost or delayed packets.

Precision Required	Example Weapons
High	Shock Rifle, Link Gun, Lightning Gun
Medium	Assault Rifle, Mini-gun, Bio Rifle
Low	Flak Cannon, Rocket Launcher, Redeemer, Ion Painter
Other	Shield Gun, Translocator, Ball Launcher

Table 1: UT2003 Weapon Precision

The "high precision" weapons require timely response in order to accurately determine an opponent's location while aiming and may be the most impacted by lost or delayed packets.

The "medium precision" weapons are less accurate and so do not require as accurate an opponents' location while aiming and may be less impacted by lost or delayed packets.

The "low precision" weapons require merely aiming in the general direction of the target in order to hit without needing precise location information, and therefore may be the least impacted by lost or delayed packets.

The "other" weapons are not commonly used, or serve special purposes within particular types of games and so we do not consider them further.

In addition to the numerous maps, weapons and game play modes, UT2003 also comes standard with two more features which we found useful in our experiments: bots and mutators. *Bots* are computer controlled players that run on the server, each with their own personality and style of play. *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 only the Shock Rifle and making it so that every shot instantly kills an opponent.

3. APPROACH

In order to empirically measure the effects of latency and loss on Unreal Tournament 2003 (UT2003), we employed the following methodology:

- Categorize user interactions in typical FPS games and design maps that exercise each type of interaction (see Section 3.1).
- Construct a test environment for measuring the effects of latency and loss on UT2003 (see Section 3.2).
- Conduct pilot studies to determine realistic ranges for packet loss rates and latencies to characterize typical UT2003 network turbulence (see Section 3.3).
- Conduct numerous user studies on our maps inducing the appropriate range for loss and latency using our test environment (see Section 3.4).
- Analyze the results (see Section 4).

3.1 Categories of FPS Interaction

Through play testing, we determined there are two main user interaction components in FPS games: *movement* in which users navigate through the game map from one location to another; and *shooting* in which users aim and shoot weapons at opponents. Further study of movement suggested that simple movement, such as running as quickly as possible in a straight line towards a goal, is fundamentally different in the amount of interaction than complex movement such as jumping, dodging, and navigating obstacles. We hypothesize network degradation affects complex movement more than simple movement. Further study of shooting suggested that aiming depends upon the weapon's precision (see Table 1 in Section 2). We hypothesize that network degradation affects precision shooting more than normal shooting.

Our movement test maps consisted of running pre-defined routes in regular game maps. The simple movement test, based on the Tokara Forest map included in a standard game install, had a player run in a straight line. The complex movement test,[4] based on the standard Flux2 map, had a player run, jump, spin and pickup items in an obstacle course.

Our precision shooting test map, based on the standard CTF-Face3 map, featured one player aiming and shooting a high precision weapon (the Lighting gun) at a second player from a distance while the second player tried to dodge to avoid being hit. Since the use of less precise weapons invariably involves movement combined with shooting, we designed another map, based on the standard Training Day map but without health bonuses, that pitted one player against a bot ("Widowmaker") where both player and bot had their weapon choices limited to medium precision weapons using the insta-gib mutator. This forced the user to aim and dodge concurrently because bullets did not have a wide spread and there was no way to gain life to overcome damage from the bot. There were enough obstacles for cover and to prevent an individual from gaining too much of an advantage from spawn camping.[5] It should be noted that since the bot runs on the server, it does not suffer when network conditions are degraded, but any other human player does.

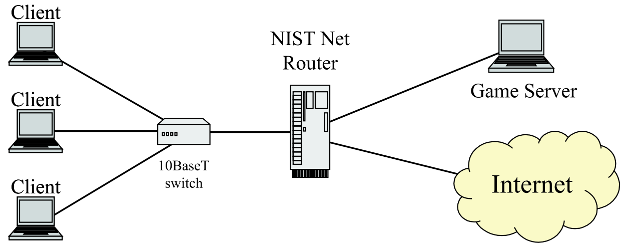

Figure 1: Experimental Testbed Setup.

Lastly, we used the standard Training Day map to study full length games with a normal array of weapon choices in order to study the full interaction of movement and shooting. As in the previous map, the game was limited to two players, one being a human and the other a computer controlled Widowmaker bot.

3.2 Experimental Environment

We designed a lab in which we could systematically control loss and latency while running out custom test maps. Figure 1 shows our testbed setup. Our lab had four client computers running the latest version of Unreal Tournament 2003 (v2225). The four clients were connected to a 10 Mbps 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 UT2003 as a dedicated server. The third network card connected directly to another computer configured to act as a gateway to the WPI network[6] and the Internet.

For tools, we used the NIST Net[7] network emulator on the router between the server and the clients in order to provide fine-grained control of packet loss rates and latency for individual clients. We used Ethereal[8] on the router to capture all packets traveling between the clients and the server. We used the All Seeing Eye[9] to gather UT2003 Internet server statistics including packet loss rates and latencies which provided the basis for the values used in our experiments.

3.3 Pilot Studies

We first ran network packet traces of several UT2003 games in order to observe "standard" network traffic we could later compare to UT2003 network traffic in the presence of loss and latency. We also used the All Seeing Eye (ASE) run from both WPI and a local DSL connection to ascertain appropriate packet loss and latency ranges for study.

Based on the data from the ASE,[10] about 80% of all UT2003 game servers have no measurable loss, with only 0.1% having loss rates above 2.75%, and the maximum reported loss rate is just over 3%. About 40% of all UT2003 game servers have latencies of less than 100 ms, another 40% are between 100 and 140 ms, and only 20% of all servers ex-

[4] See [2] for a detailed specification of the test, including a screen-shot walk-through.

[5] *Spawn camping* is when a player waits near the location where an opposing player will come back to life for a quick kill.

[6] See http://www.wpi.edu/Admin/Netops/MRTG/ for WPI's network setup.

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

[8] http://www.ethereal.com/

[9] http://www.udpsoft.com/eye/

[10] Graphs of the cumulative distribution functions for the data gathered with ASE can be found in chapter 6 of [2].

hibit latencies greater than 140 ms. Based on this pilot study data, we explore loss rates in the range [0%,6%] and latencies in the range [0ms, 400ms] for our experiments.

3.4 User Studies

Over a period of about one month, we ran over 200 experiments representing hours of FPS game play. All experiments were conducted on our testbed using our pre-defined maps. Although we did not quantify user skill levels, all users were very familiar with UT2003.

Our methodology consisted of allowing the users to familiarize themselves with the game play map with no loss and latency before collecting any data. Then, the experiment operator would induce an amount of loss or latency selected from the experimental range and run the experiment. The users were thus "blind" to the amount of loss or latency in order to avoid having knowledge of the network conditions bias their play. After the experiment was completed, the operator would archive the data for later analysis, modify the amount of loss or latency, and the users would repeat the experiment.

4. ANALYSIS

We analyzed our experimental data at three levels: Section 4.1 contains our analysis of the application level data that we collected from our UT2003 user studies; Section 4.2 presents our analysis of the network level traffic for full games with three levels of induced latency; and Section 4.3 summarizes the subjective observation data we collected during the user studies.

4.1 Application Level Analysis

This section analyzes the results from each of our test maps: movement (Section 4.1.1), precision shooting (Section 4.1.2), restricted game play (Section 4.1.3), and full game play (Section 4.1.4).

4.1.1 Movement

A player's ability to move his or her avatar around a game map is a critical aspect of a FPS (and most computer games, for that matter). So while the primary goal of our movement tests was to determine the impact of network degradation on player movement, another was to determine how the UT2003 handles delayed or dropped packets in relation to a player's movements, possibly through various latency compensation techniques [3, 14]. Such techniques typically reduce the impact of latency and perhaps ignore some packet loss at the expense of data consistency for all clients.

Simple Movement

In the simple movement test, a third player stood on the side and determined by observation the winner of a footrace between two other players. The footrace test was conducted with multiple experimental runs, with each run having a different amount of induced loss or latency. For all experiments, the player with added network loss or latency and the player without added loss or latency crossed the finish line at about the same time.

From these results, we conclude that packet loss and latency do not have any measurable effect on an avatar's raw speed. We can also infer that calculation of a player's location requires minimal interaction from the server and thus UT2003 uses some form of client-side prediction for latency (and loss) compensation.

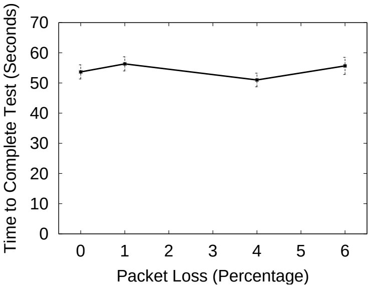

Figure 2: Complex Movement Test - Time to Complete versus Packet Loss.

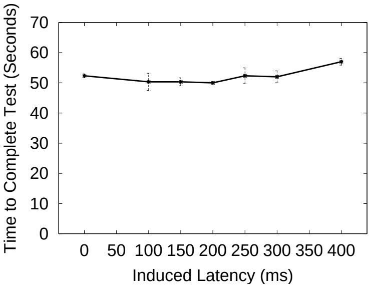

Figure 3: Complex Movement Test - Time to Complete versus Latency.

Complex Movement

For the complex movement test, we recorded the time it took users to navigate the obstacle course, repeating the test over a range of loss and latency values. Figure 2 illustrates the average of the three test times for different loss rates with the mean points shown with 95% confidence intervals. From the figure, packet loss has no noticeable effect on course completion times.

Figure 3 shows the average course completion times and 95% confidence intervals of three test times for different induced latencies. From the figure, latency has no noticeable effect on course completion times up to latencies of 300 ms, after which a slight upward trend can be seen that continues through the tests to 400 ms. Although the slight upswing perhaps suggests some correlation with complex movement and latency, our server ping time statistics in Section 3.3 have shown that servers with latencies above 300 ms are infrequent.

Movement Summary

The results of these tests indicate that neither latency nor loss have a noticeable impact on a player's ability to move effectively in the UT2003 game environment. We surmise this lack of impact is primarily due to a short-circuit relay that allows the client to initiate and update a move locally and inform the server (and other players) at a later time, thus allowing smooth performance in the presence of network degradation.

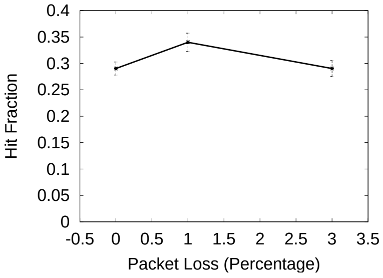

Figure 4: Precision Shooting - Hit Fraction versus Packet Loss.

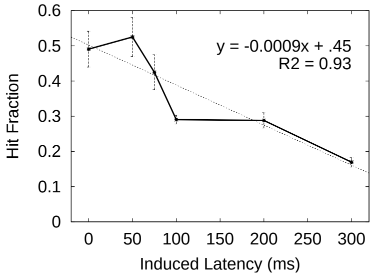

Figure 5: Precision Shooting - Hit Fraction versus Latency.

4.1.2 Precision Shooting

For the precision shooting tests, we recorded the fraction of hits in a 10 minute game at each loss and latency using our precision shooting map. The loss tests also included a baseline latency of 100 ms in an attempt to emulate a more realistic Internet game. Each experimental run was repeated 3 times by each of 2 players with comparable skill. Figure 4 depicts the mean hit fraction shown with 95% confidence intervals. While the confidence intervals are non-overlapping for the case of 1% loss, the confidence intervals overlap for the 0% and 3% cases and the means reside inside the intervals. Thus, we conclude packet loss does not have significant impact on precision shooting performance.

We repeated our experiments with a base loss rate of 0% and a range of latencies induced on the person firing. Again, we ran tests with 2 different players of comparable skill with a total of 3 times for each player at each induced latency amount. Figure 5 depicts the mean hit fraction shown with 95% confidence intervals. While there is a slight downward trend in the hit ratio for latencies up to 75 ms, the overlapping confidence intervals covering the means indicate the differences are not statistically significant. However, at 100 ms there is a sharp change where the mean accuracy drops to approximately 0.33, down about 35% from the mean accuracy with less latency. As latencies increase above 100 ms, shot accuracy continues to decline further, with a decrease of over 50% at a latency of 300 ms. The confidence intervals for the mean hit ratios with latencies of 100 ms or more do not overlap, indicating statistically significant differences. The linear regression in the figure illustrates the downward trend as latency increases and the coefficient of determination[11] is a high 0.93.

Precision Shooting Summary

Precision shooting is robust enough to not be affected by even the most extreme packet loss found in typical UT2003 games. We assume this is because the number of packets containing shot data represents a small percentage of all transmitted packets and so does not affect a player's ability to aim with precision. However, precision shooting is very sensitive to latency, with a steady decrease in hit accuracy for latencies of 100 ms or over.

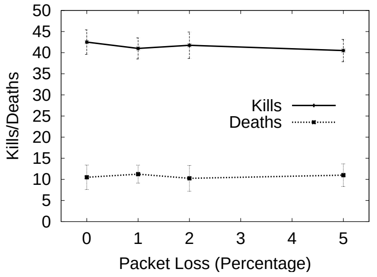

Figure 6: Restricted Deathmatch - Kills/Deaths versus Packet Loss.

4.1.3 Restricted Deathmatch

After completing the experiments that independently tested the effects of shot accuracy and movement, we next conducted tests that combined shooting and movement in our restricted Deathmatch map. After each 5-minute match using the map, we recorded the number of kills and deaths accumulated by the human player. We tested four users at each loss and latency amount. Figure 6 shows the mean number of kills and deaths at each packet loss rate with 95% confidence intervals around each mean. From the figure, the packet loss made no noticeable difference in user performance as the confidence intervals overlap all the means.

Figure 7 shows the mean number of kills and deaths at each latency amount with 95% confidence intervals around each mean. There is a visual trend that shows a decrease in player performance as latency increases. Up to 75 ms of induced latency the confidence intervals overlap and include the means, but the kill average at 100 ms of latency does not reside in the confidence interval for the kill average at 0 ms latency. Similarly, the death average at 75 ms of latency does not reside in the confidence interval for the death average at 0 ms latency. Similar statistical statements can be made for the kill averages with 150 ms and 200 ms compared with 100 ms. The linear regressions for kills versus latency has a high coefficient of determination of 0.92 and the linear regression for deaths versus latency has a modest coefficient of determination of 0.73. It may be that deaths are slightly less affected by latency than are kills because the act of dodging is not as demanding in terms of response time as is aiming and shooting.

[11] The coefficient of determination (R^2) represents the fraction of variability in y that can be explained by the variability in x. In the simple linear regression case, R^2 is simply the square of the correlation coefficient. An R^2 of 1 represents perfect correlation.

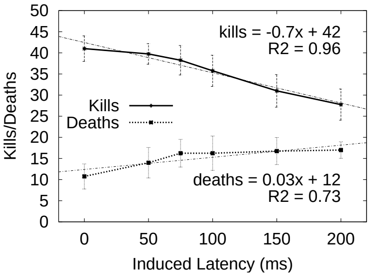

Figure 7: Restricted Deathmatch - Kills/Deaths versus Latency.

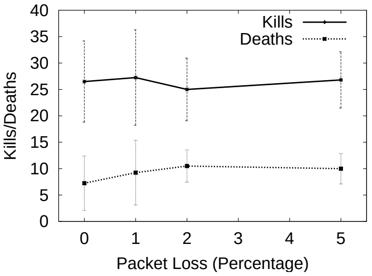

Figure 8: Full Deathmatch - Kills/Deaths versus Packet Loss.

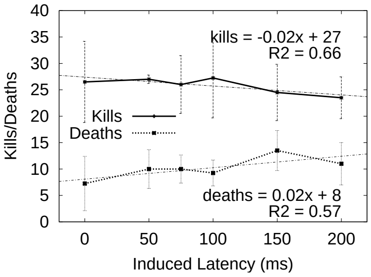

Figure 9: Full Deathmatch - Kills/Deaths versus Latency.

Restricted Game Summary

As with previous aiming and movement tests, we found that packet loss has no noticeable impact on player performance most likely because a player's reflexes under loss are not impaired. The fast pace of UT2003 makes it unlikely a small percentage of dropped packets significantly impacts performance. However, latencies of 100 ms or more cause both the number of kills to drop and number of deaths to increase, with a performance degradation of about 30% at latencies of 200 ms.

4.1.4 Full Deathmatch

In addition to isolated interaction component analysis and some limited combined analysis, we studied the impact of packet loss and latency on player performance in a regular game using the Training Day map pitting a human against a bot. We ran 5-minute matches, four times at each loss and latency level, after which we recorded the number of kills and deaths accumulated by the human player. Figure 8 shows the average number of kills and deaths at each loss amount with 95% confidence intervals around each mean. All confidence intervals overlap so there is no statistical difference between the different loss amounts, and the observed trend is a flat line. As in the previous tests, we find packet loss has no measurable impact on real world game-play.

Figure 9 shows the mean number of kills and deaths at each latency amount with 95% confidence intervals around each mean. Compared with the game with restricted weaponry, latency has only a limited statistical impact on player performance since most confidence intervals overlap and the mean values lie within the confidence intervals of each other. Still, there are apparent visual trends that indicate a decrease in performance with an increase in latency. The linear regressions for both kills versus latency and deaths versus latency have weak coefficients of determination, around 0.6.

Full Game Summary

In the full game tests, packet loss still does not impact player performance. Statistically, latency does not have a significant impact on performance either, although there is a slight decreasing trend in player performance with an increase in latency. These results may be because players can compensate for high latencies by purposely switching to weapons that require less accurate aiming. Certain weapons such as the Flak Cannon, for example, fire in a cone shape that spreads out as it travels away from the shooter. Even with a high amount of latency the player only needs to aim in the general direction of an opponent to cause a substantial amount of damage. Future work may look at how players change their strategies, perhaps choosing such low precision weaponry, at higher latencies.

4.2 Network Level Analysis

Among other things, a better understanding of network game traffic can help design networks and architectures that more effectively accommodate network game traffic footprints. Furthermore, careful empirical measurements of network games can provide the data required for accurate simulations, a typical tool for evaluating network research. This section describes and summarizes the results of our network traffic analysis of UT2003.

4.2.1 Traffic

We ran multiple full-length games with one player matched against two bots on a small standard map (DM-GAEL) with four different conditions of packet loss and latency. For each game, we captured all network packets for 120 seconds during the middle of the 5 minute match.[12]

Figure 10 shows the bitrates averaged every 500 ms over time. Visually, it appears that neither packet loss nor latency has a significant impact on a UT2003 game's bitrate. Table 2 tabulates the games' mean bitrates and standard deviations. Once again, latency and packet loss have little effect and all four traces have very low bitrates that can easily be achieved with typical access link bandwidths.

Figure 11 shows a cumulative distribution function of packet

[12] The network traces can be downloaded from http://perform.wpi.edu/downloads/#ut2003.

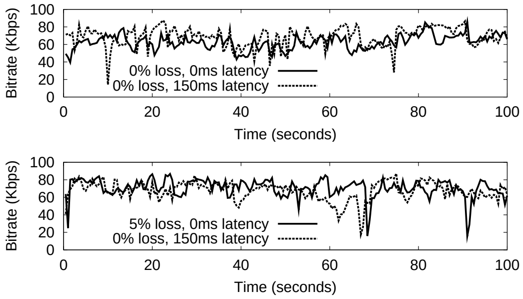

Figure 10: Bitrate versus Time for Different Network Conditions.

Loss	Latency	Mean Kbps	Std Dev
0%	0ms	63.15	9.33
0%	150ms	67.12	11.90
5%	0ms	69.87	10.86
5%	150ms	66.24	11.22

Table 2: Mean Bitrate and Standard Deviation for Different Network Conditions

sizes, including UDP/IP header plus game data, during a typical game of UT2003. Higher levels of packet loss and latency did not appear to have a significant effect on the size of the packets. In general, UT2003 packet sizes are significantly larger than a popular real-time strategy game [13], but comparable to other FPS games. Overall, UT2003 sends considerably smaller packets than the typical Internet traffic packet size of over 400 bytes [9].

Figure 12 depicts the cumulative distribution functions (CDFs) for inter-packet times, both client to server and server to client. The graphs show that the client is somewhat inconsistent and sends packets every 10-20 ms, probably on the frequency of user actions, while the server is consistent and sends packets every 50 ms, with a fewer inter-packet times of approximately twice that at 100 ms.

4.3 User Level Analysis

While we did not provide a way to quantify player perceptions, we did note game player comments and observed trends during and after our user studies.

Players were able to notice sluggishness in game play when latencies as low as 75 ms were induced on their connection, and found game play less enjoyable at latencies over 100 ms. This relationship held even for full games, and players felt they were playing poorly even if their scores were not statistically worse.

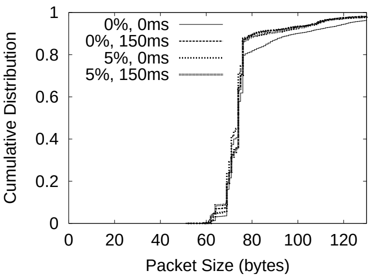

Figure 11: Cumulative Distribution Functions of Packet Sizes for Different Network Conditions

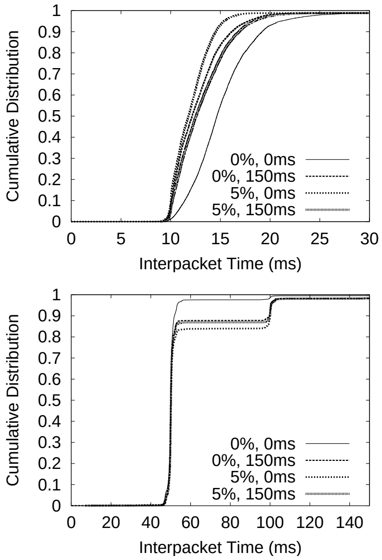

Figure 12: Cumulative Distribution Functions of Inter-Packet Times for Different Network Conditions. The top graph is Client to Server and the bottom graph is Server to Client.

Occasionally players were also able to notice packet loss when induced loss rates were at least 3%, with the primary artifact noticed being that the game would sometimes not display animations for shots that were fired. 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 most subjective impact was during the precise shooting tests. Players were extremely aggravated when trying to aim and shoot when latencies higher than 100 ms 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, particularly for the full game tests. Again, players felt as if they were performing worse, even if though their scores did not reflect it.

Generally, we recommend that players avoid servers with ping times over 150 ms and packet loss levels over 3%. Even though servers that do not meet these criteria may still not significantly impact player scores, they do make game-play less enjoyable, which at least partially defeats the purpose of playing games in the first place.

5. CONCLUSIONS

Understanding the effect of loss and latency on application performance is important in order to design more efficient networks. The growth in interactive network games demands better understanding of the effects of latency on user performance in network games. First Person Shooter (FPS) games, in particular, tend to be more sensitive to changes in network performance. Latency changes and lost packets could mean that a target is no longer where expected or bullets may never even fire.

In this paper, we investigated the effects of loss and latency on user performance for Unreal Tournament 2003 (UT-2003), a popular FPS game. We divided UT2003 into the fundamental user interaction components of movement and shooting, sub-dividing movement up into simple and complex and sub-dividing shooting based on the precision of the weapons being shot. We designed maps that allowed isolation of each component and setup a testbed that allowed a systematic control of both packet loss and latency.

Through numerous user studies, we find that packet loss has no measurable affect on player performance in any user interaction category. Moreover, users rarely even notice packet losses even as high as 5% during a typical network game. Latency has no measurable effect on simple, straight-line or more complex movements. Shooting, however, is greatly affected by latency with even modest (75-100 ms) amounts of latency, decreasing accuracy and number of kills by up to 50% over a common Internet latency ranges. While combinations of movement and shooting somewhat mask the effects of latency on user performance, even unrestricted games show trends which indicate that latency degrades user performance. This is reflected in the subjective comments collected during our user studies in which loss rates went unnoticed, but latencies as low as 100 ms were noticeable and latencies around 200 ms were annoying.

At the network level, UT2003 games basically produce small, regularly-spaced packets and modest aggregate bitrates which make it suitable for play over low-capacity devices. In fact, the bitrates make it playable over modems but the added latency caused by typical modems[7] may seriously degrade game play. Broadband access networks, however, are perfectly suitable for good UT2003 performance, both in terms delay and capacity. The network turbulence, in terms of packets and packet spacing, does not measurably change with changes in loss or latency.

UT2003, and we suspect other FPS games, would clearly tolerate modest amounts of packet loss in order to preserve low latencies. This does not bode well for mechanisms that rely upon applications to voluntarily throttle back their data rate in the presence of packet loss, but does provide promise for mechanisms that allow explicit tradeoff of higher packet loss for reduced latencies.

6. FUTURE WORK

While the results in this project have focused on UT2003, we assume that they generalize to other FPS games (such as Counter-strike[13] or Battlefield 1942[14]) as well, since most FPS games have the same fundamental components (movement, shooting and combinations) and similar user-interaction models. Studies to confirm this through select user-studies would be useful to verify our assumption and help generalize our results.

The performance of interactive applications often degrades significantly under variance in latency (or jitter), in addition to latency. We hypothesize that varying latencies would make it especially difficult for precision shooting as compensation for the perceived latency becomes difficult. Systematic study of the effects of variance in latency is a possible area of future work.

The data showing that weapons with less precision are less affected by latencies suggests users might adapt, knowingly or not, to higher latencies by choosing to fight with weapons that need less precision. Future studies that examine user strategies in weapon selection for a range of latencies may help understand how users themselves adapt their style of play and strategy to degraded network conditions.

7. REFERENCES

[1] G. Armitage. An Experimental Estimation of Latency Sensitivity in Multiplayer Quake 3. *International Conference on Networks (ICON)*, Sept. 2003.

[2] T. Beigbeder, R. Coughlan, C. Lusher, and J. Plunkett. The Effects of Packet Loss and Latency on Player Performance in Unreal Tournament 2003. *Major Qualifying Project MQP-MLC-NG03*, WPI, May 2004. Advisors M. Claypool and E. Agu.

[3] Y. W. Bernier. Latency Compensating Methods in Client/Server In-game Protocol Design and Optimization. *Game Developers Conference*, Feb. 2001.

[4] P. Bettner and M. Terrano. 1500 Archers on a 28.8: Network Programming in Age of Empires and Beyond. *Gamasutra*, Mar. 2001.

[5] W. Feng, F. Chang, W. Feng, and J. Walpole. Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server. *ACM SIGCOMM Internet Measurement Workshop*, Nov. 2002.

[6] T. Henderson. Latency and User Behaviour on a Multiplayer Game Server. *COST Workshop (NGC)*, Nov. 2001.

[7] T. Jehaes, D. Vleeschauwer, T. Coppens, B. Doorselaer, E. Deckers, W. Naudts, K. Spruyt, and R. Smets. Access Network Delay in Networked Games. *ACM NetGames*, May 2003.

[8] P. Lincroft. The Internet Sucks: Or, What I Learned Coding X-Wing vs. Tie Fighter. *Gamasutra*, Sept. 1999.

[9] S. McCreary and k claffy. Trends in Wide Area IP Traffic Patterns: A View from Ames Internet Exchange. *ITC Measurement and Modeling of IP Traffic*, Sept. 2000.

[10] Y.-S. Ng. Designing Fast-Action Games for the Internet. *Gamasutra*, Sept. 1997.

[11] L. Pantel and L. C. Wolf. On the Impact of Delay on Real-Time Multiplayer Games. *Network and Operating Systems Support for Digital Audio and Video*, May 2002.

[12] C. Schaefer, T. Enderes, H. Ritter, and M. Zitterbart. Subjective Quality Assessment for Multiplayer Real-Time Games. *ACM NetGames*, Apr. 2002.

[13] N. Sheldon, E. Girard, S. Borg, M. Claypool, and E. Agu. The Effect of Latency on User Performance in Warcraft III. *ACM NetGames*, May 2003.

[14] J. Smed, T. Kaukoranta, and H. Hakonen. Aspects of Networking in Multiplayer Computer Games. *The Electronic Library*, 20(2):87–97, 2002.

[13] http://www.counter-strike.net

[14] http://www.eagames.com/official/battlefield/1942/us/