Validation of a Taxonomy for Player Actions with Latency and Network Games

An Interactive Qualifying Project Report

By

Marco Duran, Thomas Meehan, Matt Thompson

Advisor Mark Claypool

March 2016

WPI

Table Of Contents

Abstract

As online connectivity becomes more important to interactive media, dealing with network latency has become increasingly important. This project was designed to study the validity of a taxonomy to classify the impact of player actions with latency. We utilized a commercial game to simulate latency in a first person shooter match where participants competed against a computer controlled opponent. The participants utilized three different weapons: a shotgun, a rocket launcher, and a sniper rifle. Each weapon was designed to embody different characteristics of the taxonomy axes: precision, impact, and deadline. The participants used each weapon against the opponent at differing amounts of latency. Analysis of the data shows that lower precision weapons like the rocket launchers and shotguns were significantly less accurate when latency was increased, while high precision weapons such as sniper rifles dealt less damage as latency increased. Overall, we partially confirmed the validity of a previous taxonomy. Our findings fit the taxonomy in regards to the impact of damage and the weapon's shooting speed on a player's performance but the results were inconclusive on other aspects of player actions. Future research would benefit from utilizing a much larger, more skilled participant pool, and finding or creating software that more robustly simulates playing games online with varying levels of latency.

1. Introduction

Latency, in its most general form, is the time interval between an action taking place and its effects becoming apparent in the observed system. In the context of this paper, latency generally manifests itself over a network where commands input by one user have to travel across the network before reaching a central server or other users. In the context of games, this can lead to multiple conflicting game states between the various clients and servers. It is important to note that latency can and often does vary enormously between observers, even with regards to the same action. Often, different distances from the clients to the server is the most direct reason for differing levels of latency. After all, even the most efficient transmission of information is limited by the speed of light, so there is always an absolute limit on how fast data can be transmitted based on the distance between the two points. Despite this limitation, there are ways latency can be reduced and its apparent effects minimized from the user's perspective. To compensate for latency, we must first consider why latency is such an important issue and why it persists despite recent advancements in high speed communications.

Regarding the exact causes of network latency, on top of simply having to send information across great distances, there are many other factors to consider, such as having different subnetworks or dealing with sending info to multiple clients at once. Communications must travel through many levels of subnetworks, ranging from local ISPs, to city mainlines, to national communication links, and sometimes international networks. Every level encountered and subsequent transition between them adds additional processing time to the transition, ultimately increasing the time it takes to transmit packets of information, thus increasing latency. Furthemore, each subnetwork serves as a potential traffic bottleneck, such as a broadband neighborhood headend having to deal with several high-traffic clients simultaneously, resulting in high latency. It is also important to note that not all the information is sent all at once, and that pieces of it can get lost along the way. Some network traffic may never make it to its destination, further exacerbating the effects of latency by requiring lost traffic to be re-sent. Finally, while many networks use high speed components such as fiber optic cables, the hardware connecting to individual homes is often much slower, drastically increasing latency in what is commonly referred to as “the last mile” problem.

As for how latency manifests itself in network games, on the most basic level the game has to make assumptions on the current game state until it gets its next update from other participants (and by the time the update arrives it is already somewhat out of date). While games do their best to predict the current game state using a variety of techniques, predicting individual player behavior can be tough, and many issues arise when predictions are finally reconciled. The most common issue is that player avatars appear to teleport from one location to the next since the local game client believes that the player was originally in one place, but the player was actually in a different location. This can also lead to frustrating experiences where players will perform actions based on where the local client believes the opposing player is, but in reality the opposing player is somewhere else. For example, there are moments where players will shoot opponents with no effect, or where player characters will appear to spontaneously die behind cover due to their opponent’s local client believing that the player was still exposed and thus vulnerable to attack even though the player had actually moved to cover. This behavior often leads to players gravitating toward more reliable means of attack, for example melee strikes in a combat game, which tend to have a much larger margin of error in high latency environments due to not having to aim to hit your opponent. Nevertheless, even these more reliable actions can still be highly affected by latency. Due to the nature of melee attacks often being a one-hit kill, there are many instances where players will ‘kill’ their opponent, but the latency results in that the opposing player remaining alive much longer, often giving them ample time to strike the other player in turn. As a result, it's possible to see two players punch each other, then both collapse immediately after. On the other end of the spectrum, long time frame attacks such as grenades that go off after a couple of seconds may appear out of nowhere with half a second left on their timer instead of the intended 4-5 second window, leading to player frustration when they are given no time to react to a certain death attack.

In order to help address the latency issue, a previously proposed taxonomy showed the effects of player performance when prioritizing player actions [1]. The original taxonomy primarily focused on incorporating a way to measure player performance with network latency along two axes; precision and deadline. Deadline is the length of time it takes to execute the outcome of the action. Precision is the accuracy required to complete the action with success. Since these axes are dependent on the impact of player actions on the game world, there is not sufficient information to quantify the data collected. The revised taxonomy deemed that a third dimension, player impact, is necessary to acquire quantifiable data based off of player actions.

A previous study [1] used a custom arcade shooter game to look at the different dimensions of the taxonomy. The results of the experiment show that the expected impact, or the damage the player does, is greater when the damage of the player's weapon is greater. The expected impact also increases when the speed of the player's weapon is increased. When dealing with area of effect actions, the results of the study show that the expected impact also increases when the area of effect is greater. Overall, the results of the previous taxonomy show that the increase of network latency correlates to a decrease in expected impact that the player has on the game world.

As previously noted, the evaluation of the original taxonomy was fairly limited. It only tested the dimensions using a bot, in a 2d text-based game. Our goal is to validate it by testing a different game type, with actual players for our tests. The use of human testers should result in more conclusive data, as the main objective we are testing is the effect of latency on player actions.

For our project, we will be conducting our experiments with a First Person Shooter (FPS) game. The main reasons for choosing an FPS are as follows. Firstly, most gamers have played an FPS before, so finding experienced participants should be fairly easy. Secondly, performing well in an FPS requires Precision and Deadline, which make it a prime candidate for validating the taxonomy. Thirdly, we need a free to play game with good documentation for our testing, and FPS's can provide us with several choices.

We used the game Team Fortress 2 (Valve Corporation, 2007, http://www.teamfortress.com/) for our testing. With it, we controlled everything in our tests from the level design to the weapons. We made our own map in Valve's Hammer World Editor. We used Source Engine's (Valve Corporation, 2004, http://www.valvesoftware.com) built-in network modifier to simulate latency and used custom SourceMod plugins to limit the player's weapon, class and the opponent's class. We solicited users by posting advertisements on campus, emailing computer science and game development majors, and submitting our study to the psych participant pool.

We administered multiple surveys to our participants: one pre-test survey and a few post-test surveys. The main point of the pre-test survey is to evaluate the level of experience the player has with FPS's and to gather demographic info. This can help us put our data into context. Our post-test survey mainly asked questions about how the player felt during the test.

For example, we wanted to know if they felt that they had an advantage over the other player. We administered one post-test survey per different test per participant.

In total, we had 41 participants. Each participant played 4 matches with each class; one for each level of network latency we were testing. We tested with network latency values of 0ms, 100ms, 300ms and 500ms.

We mainly examined three statistics from the user study: average accuracy, average win rate and average damage dealt in one round. We found that when using the Shotgun and the Rocket Launcher, the player's accuracy decreased overall as latency was applied. On the other hand, when using the Sniper Rifle, accuracy actually increased. Overall, the average win rate increased for each weapon as network latency was applied. On average, the participants dealt less damage per round with the Sniper Rifle as network latency was applied. The average damage dealt with the Shotgun and the Rocket Launcher, however, increases as latency was applied.

The remainder of the paper is structured as follows: Section 2 discusses related works that were studied prior to conducting our own study; Section 3 contains all of the information relating to the game that we used for the study; Section 4 outlines the methodology for our experiment; Section 5 shows all of the data that was collected from the study; Section 6 contains a discussion of the findings from Section 5; Section 7 is the conclusion of the project.

2. Related Work

While there have been many studies regarding how latency affects player performance within video games, many of them focused only on overall impact within a specific genre of game, and did not delve into how specific player actions were affected. Such studies have included online role playing games [12], real time strategy games [7], racing games [11], and first person shooters [2]. These studies examined the impact latency had on the games as a whole, and were ultimately only able to apply their findings to their respective genres, but nevertheless proved useful in helping researchers better understand the impact of network latency within common game genres.

The recent emergence of cloud-based game services has led to several modern studies investigating the effects of latency on completely network based gaming platforms. Due to the nature of these platforms, where both input and response are limited by network performance even during gameplay that could otherwise be handled completely locally, the issue of how latency affects performance and player experience was of particular interest to these studies. Some studies focused mostly on the the overall responsiveness of cloud based platforms [14, 15] while others conducted user studies that focused on observing the effects the platforms had on the player [16, 17]. However, much like aforementioned studies on traditional games within genres, these games did not isolate the effects of latency in terms of specific player actions.

On the other hand, there are some studies that have focused specifically on isolating player actions within games and how latency can affect various player actions differently than others, such as combat in real time strategy games [9]. This eventually lead to attempts to classify player actions in terms of latency sensitivity [18], leading to the creation of the taxonomy this paper hopes to validate [1].

One study in the field of network latency in games has looked at overall player performance and the fairness of experiencing latency during a competitive match [4]. The results of this study show that the fairness between two players is dependent on network latency. More specifically; network latency of 30 milliseconds and above results in unfairness between players. Relating to the fairness that players have in a game with latency, another study has looked at different ways to manage playability and fairness within network games [8]. In this study, player fairness in network games is determined by standard deviation of server response time. Playability is measured as the average response time of the server. Using these values, the study confirmed that a smaller response time from the server results in a more fair and playable experience for the user. The study states that managing latency and fairness in network games is difficult to accomplish on a large scale. In our study, we plan on studying different player actions with network latency in order to determine if it is possible to view the effects, rather than strictly observing the server.

Other studies have simply focused on pure player performance during the tests. In a recent study on the impact of latency on player performance in Unreal Tournament [2], the researchers studied the effects of packet loss and latency on user performance in Unreal Tournament. They found that packet loss has little to no effect, and that latency effects shooting a lot more than moving. Two recent studies tested user performance in Real Time Strategy (RTS) games [7, 9] and are similar in that they both tested user performance in three categories: exploration, building and combat. Both studies found that although latency had a slight impact on exploration, it had little to no effect on building and combat. They also found that although the players noticed latency in their games, it did not affect the outcome of the game. They concluded that this is most likely because of the emphasis of overall strategy in RTSs, as opposed to the emphasis of reaction times and quick movements in other games.

Although testing player performance is useful, it is fairly limited as many factors go into individual player performance. Our goal is to test each player action, as defined by the taxonomy, to more effectively judge the effect of network latency on player performance.

3. Background

In this chapter we will explain how Team Fortress 2 was selected and describe the characteristics that are utilized in our research. We will also explain the different classes that were used in the game as well as information pertaining to how the map for the study was created. We have selected TF2 over other free-to-play first person shooter games mainly because the game has been active for nine years, the ease of access to the developer console, a range of available, legal and free modifications to the game, and a plethora of documentation to help in the setup of a custom server. Overall, these were the features and configuration options in TF2 that were used and analyzed for our research.

In our study we had participants play Team Fortress 2 (TF2). TF2 is a free to play game on Steam, an online market for games. We created two login accounts that keep track of the players during the experiment and also ensured that experienced participants had no way of using custom weapons during the study.

TF2 offers multiple weapon types and different game modes. The three classes that were used were the Scout, Soldier and the Sniper. The Scout is highly maneuverable and moves fast, but has a low amount of health. The Scout's primary weapon is a shotgun that deals large amounts of damage at close range. The Soldier moves slower than the Scout and has more health. The Soldier is equipped with a rocket launcher that deals damage in an Area Of Effect (AOE). The Sniper uses a sniper rifle that, when scoped, charges in power for a few seconds. The more the rifle is charged, the more damage the shot deals. Also, a headshot with a scoped sniper rifle results in a critical hit that deals massive amounts of damage and usually kills the opponent in one shot.

To create a level in TF2, we used Valve’s Hammer world editor. For our server creation, we used Valve’s “Source Dedicated Server” tool. This tool allowed the server to run from a computer that did not have the client version of the game. This is mainly used to host many games from one computer. The server allowed modifications to be made to the game by the installation of SourceMod. SourceMod allowed installation of community made C++ addons to help show some useful tools and completely change how the game is played.

For the user study, we enabled custom SourceMod plugins to restrict the participant’s class choice, the opponent’s class choice, to disable the latency compensation, and force a non local connection to the clients. We also used custom plugins to collect in-game statistics for each user and write them to a log file. We then wrote a statistic parser in Java to collect all of the important information from the logfile, and order it in an understandable manner. In order to properly conduct the study, there needed to be a way to modify the amount of network latency for each participant. SourceCMD’s commands let you modify the game values for TF2 to allow the server controller to change the amount of fake, or simulated, latency. The command to modify the latency is “net_fakelag ‘latency in ms’”.The ‘latency in ms’ represents the number of ms to delay. The players are able to connect to the custom server by using the TF2 client. In order to keep the map from changing when the player finishes a certain number of rounds and to disable the latency compensation for the client, we wrote a configuration script that was run when the game starts up. We also used TeamViewer, which is a remote desktop tool, to control the server machines during testing.

4. Methodology

This chapter describes how the user study with Team Fortress 2 was conducted. We go into detail about how the experiment was designed and executed. The organization of the methodology is as follows; Section 4.1 explains the creation and description of the custom map used for the user study; Section 4.2 shows how the network latency was simulated and applied to each user study; Section 4.3 gives a description of each of the modifications that were used in the user study; Section 4.4 describes the environment that was used to conduct the user study; Section 4.5 explores how we acquired participants for the user study; Section 4.6 has an explanation of the surveys that were administered to the participants. There is a description of both the survey that was given before the experiment and the survey that was given after the experiment; Section 4.7 shows the procedure of the user study and all of the steps we took to complete each study; Finally, Section 4.8 is a summary of this chapter.

4.1 The Map

The map was created with Valve's Hammer world editor. Figure 4.1-1 illustrates our map design. The map was a large arena with a barrier in the middle so that neither the player nor the bot could shoot the other as soon as the round started. The map was symmetrical, with identical spawn conditions for both sides. In our pilot studies, we noticed that the player might sometimes run out of ammo. To fix this, we put ammo pickups in each of the corners. To use the ammo pickups, the player just had to walk over one to restore their

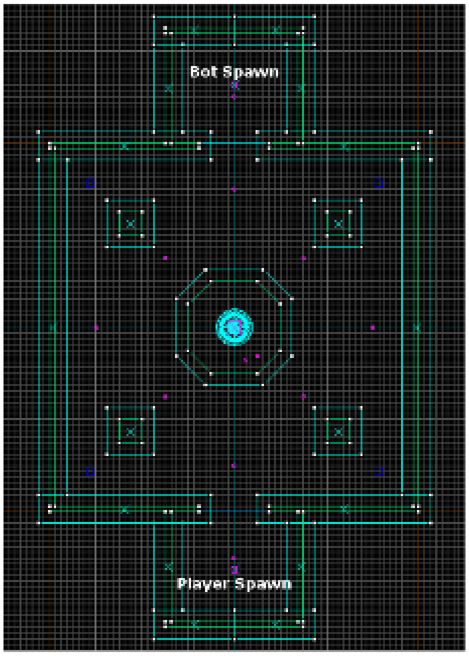


Figure 4.1-1 Overhead view of the custom map

ammo. The map had multiple edges, corners and columns that the player and bot can get behind for cover to encourage skill-based play.

4.2 Simulating Latency

After each round, we entered “net_fakelag (latency value)” into the server console. This command adds artificial latency to the game, making players experience the latency. The latency values that we used in each round were 0 ms, 100 ms, 300 ms, and 500 ms.

4.3 Mods

In order to collect the data from each participant, we modified the server with custom extensions called plugins. The only way to effectively use scripting was to install SourceMod [19] which is a library of plugins used to modify certain private variables within games created by Valve. We also turned off the variable “sv_cheats” on the server to allow other plugins with SourceMod. Since we needed to collect the statistics of each round, we chose to use a plugin called SupStats2 [20]. This plugin gave a detailed description of how many shots were fired from each weapon, how many shots hit, and how much damage was applied on each hit. This plugin also shows which player won the match, and the class of each player in the round. The plugin added all of the statistics into the logfile of the server’s console during the time an instance of the server was started. We then took each log file that had data from SupStats2 and made the data more readable by putting it through a custom file parser that we coded in Java. This parser took log files from the server logs and converted them into CSV files, which are an easier format to read and calculate statistics. In order to reduce mistakes we also used a SourceMod plugin called “TF2 Weapon Limiter” to restrict the classes that the user is allowed to pick on each round [21].

4.4 Lab Environment

The test was conducted in one of the sections of room A21 in WPI’s Fuller labs. The room has two doors, with lab computers lining each wall. We used two of the lab computers in one of the corners for the testing, and ran the servers on laptops through remote desktops. When we conducted the tests, we cleared the room and closed the doors to ensure a quiet testing environment. The participants logged into their own student accounts but we locally saved the custom Steam logins that were created for the test. The customized Steam login allows us to organize the player data with ease. The computer that the participants used had an Intel® Core^{TM} i7-4790K processor with the CPU clocked at 4.00 GHz with 16.0 gigabytes of RAM running on Windows 7 Service Pack 1 operating system. They played the game at a resolution of 1920 x 1200 on a 27 inch Dell monitor. They used headphones borrowed from WPI’s Academic Technology Center.

The servers were controlled remotely from laptops using TeamViewer. Pre-generated config files were used to swap out different server configurations depending on the weapon the participant used at the time. One researcher was in charge of monitoring each server from a laptop, while another was responsible for observing gameplay and administering pre and post surveys at the appropriate time as well as reading the script for instructing participants.

4.5 Participant Acquisition

To get our participants, we utilized two main methods. Firstly, we used direct advertisements in the forms of mass-emails to the CS and IMGD undergraduates of WPI, posts on the WPI Class of 2017, 2018, and 2019 Facebook pages and posters in most of WPI’s buildings. The second method we used was the WPI Psychology participant pool. This is where students that are currently taking a psychology class were able to elect to participate in our study in order to receive credit in their respective psychology courses.Thus, our participants consisted entirely of WPI undergraduate students. As an incentive to participate, all participants that provided contact information were automatically entered to randomly be selected to win a $30 Dunkin Donuts giftcard.

4.6 Surveys

4.6.1 Pre-Survey

Before each participant started the test, we asked them to fill out a pre-test survey so that we can better organize our data and get a better idea of the types of players in our sample pool. Getting knowledge on previous experience playing games allows us to understand how severely the latency and different experimental conditions affected their performance in regards to their own baseline, and later allows us to categorize data based on self-reported prior experience. The pre-survey questions were as follows:

*Note that the type of answer is shown in the parentheses next to each question.

List some games you have played regularly in the past 6 months. (Text)

How many hours a week do you play on average? (Multiple Choice)

- <1
- 1-5
- 6-10
- 11-20
- 20+

What genres do you play? (Checkbox)

- Action / Adventure
- Sports
- Racing
- RPG
- RTS
- FPS
- Mobile / Casual

Classify your skill level with FPS games on a game console. (1-5 Unskilled to Skilled)

Classify your skill level with FPS games on a PC. (1-5 Unskilled to Skilled)

What platforms have you played games on in the last 6 months? (Checkbox)

- PS3
- PS4
- X-Bone
- XBox 360
- PC
- Wii-U
- Vita
- 3DS
- Mobile

Have you played Team Fortress 2 before? (Yes/No)

How did you hear about this survey? (Text)

4.6.2 Post-Survey

After the tester finished a part of the test (Playing four rounds with one weapon) we gave them a post-test survey. These surveys are meant to record whether they were aware of the latency, and if so how it consciously affected their performance and enjoyment of the game. The post-test survey questions are:

How playable was the game with latency? (1-7, Unplayable to Playable)

How enjoyable was the game with latency? (1-7, Unenjoyable to Enjoyable)

Did latency affect the difficulty of the game? (Yes/No)

Did latency affect how you would normally play the game? (Yes/No)

How so? (text)

4.7 Procedure

Once all the equipment was set up we began testing with approximately 5 pilot studies. Participants were brought into one of the secluded sections of WPI Fuller A21 to use one of the lab computers adjacent to the lab servers to minimize actual latency. The same two computers were always used for ease and experimental consistency. After they are logged into the computer and TF2 is set up, participants were read an introduction explaining the basics of the study, what they would be asked to do, and how to play the game. They were then asked to practice playing the game for approximately 5 minutes under nearly identical conditions to the actual trial. During this time they were able to get accustomed to the map, controls, opponents, etc. They were also able to freely switch between any of the 3 classes / weapons being used in the study.

After 5 minutes had elapsed participants were asked to fill out a pre-survey asking the some basic demographic questions as well as questions about their experience with games (see section 4.6.1)

Once the pre-survey was complete, participants were asked to start their first official trial with one of 3 randomly chosen weapons: The shotgun, the sniper rifle, or the rocket launcher. The order was randomized for each participant in order to minimize learning bias. Once a weapon was assigned, participants had to face an AI controlled opponent in single combat. For all trials, the opponent was always a Heavy armed with a mini-gun. The round ended when either the opponent or participant was defeated. During the first round, no artificial latency was induced, leading to effectively zero network latency during gameplay. In subsequent 3 rounds artificial latency was added in amounts of 100 milliseconds, 300 milliseconds, and 500 milliseconds respectively.

After all four rounds were completed, the participant was given the first of 3 “post-surveys” where they answered a series of questions about their experience playing the past 4 rounds with a specific weapon, and how the latency affected their performance. The first question asked whether they actually noticed any latency with that weapon. If they answered yes, they were asked questions about how latency has affected their performance and enjoyment (see section 4.6.2). Otherwise they continued on to the next set of rounds for a new weapon.

Once all weapons and post surveys were completed, the participant was given the following debrief, after which the study concluded:

“This study involved observing how latency affects gameplay in different forms, specifically different actions related to deadline, precision, and impact. Each weapon featured different levels of these attributes, and we also varied the latency in each round. We ask that you do not talk about this with other participants until the study is complete. Do you have any questions?”

4.8 Summary

In summary, we customized Team Fortress 2 and had participants compete in 3 sets of 4 rounds against an AI bot. Each set the participant used a different weapon, and during each round the latency was increased substantially until the beginning of the next set. Before the study we collected basic info on their experiences playing games, and after each set we collected data on how the latency affected their gameplay.

5. Results

We organized the data that was collected from the surveys and log files and then used Microsoft Excel to generate graphs of the most relevant results. The graphs represent the demographics, average accuracy, average damage, and average win rate for each weapon across different levels of network latency. The graphs also represent the findings from a post study survey to show how the participants reacted to this study.

5.1 Demographics

Below are the results to the pre-survey multiple choice questions. Each question is in the context of gaming and is designed to gather the demographics of our sample pool.

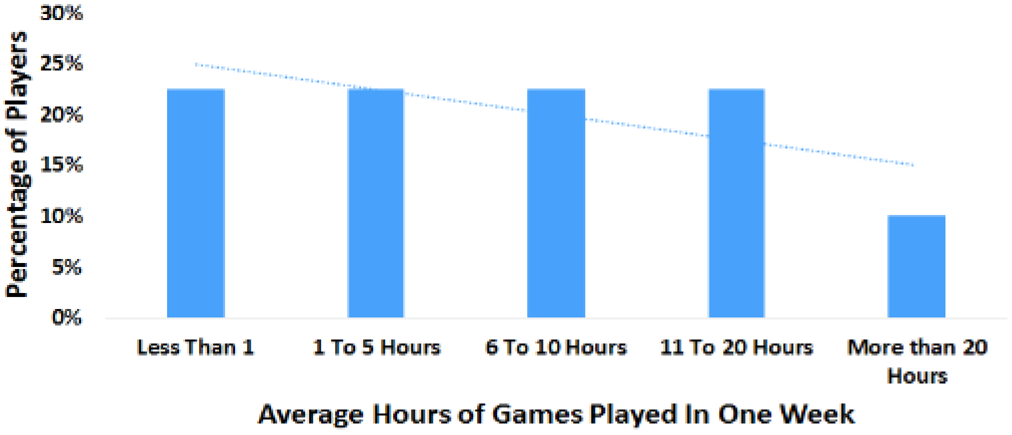


Figure 5.1-1 Average hours of games played by participants in one week

The answers to the question “How many hours a week do you play on average?” can be seen in Figure 5.1-1. A large percentage said that they played less than 20 hours a week, while only a small portion answered that they played more than 20 hours a week.

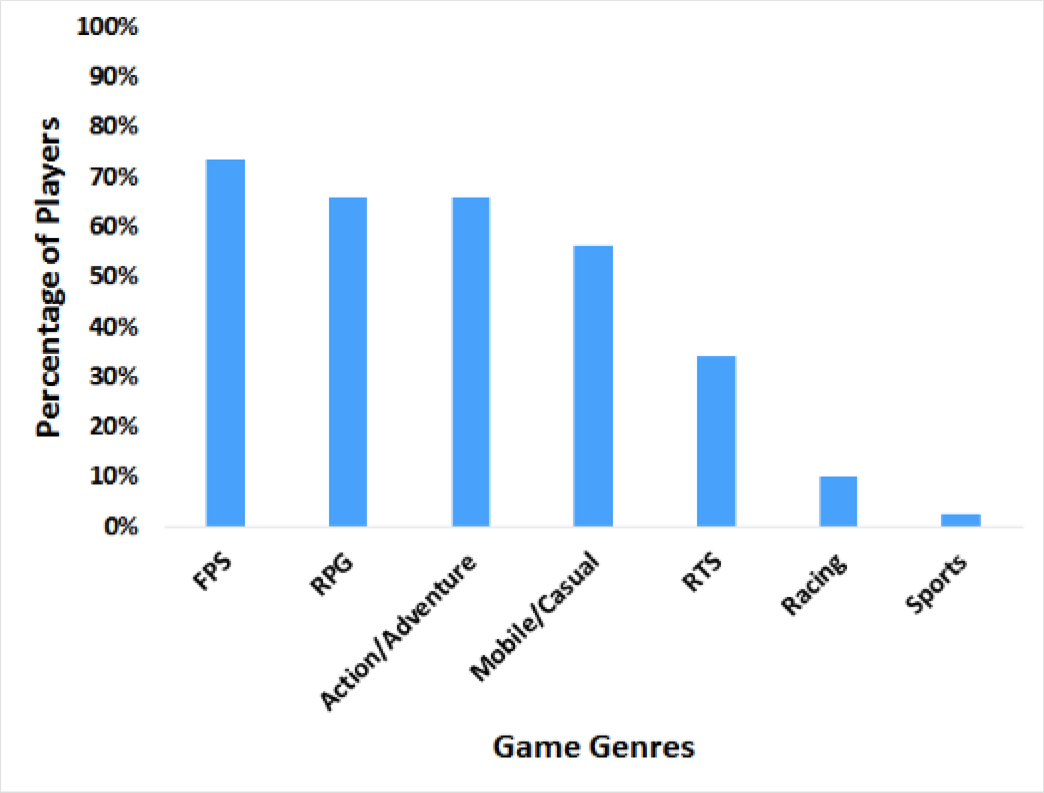


Figure 5.1-2 Game genres participants have played.

Figure 5.1-2 displays the breakdown of “What genres do you play?”. In this question, participants could give more than one answer. Most participants answered that they played FPS’s, RPG’s and/or Action Adventure games. Mobile/ casual games were also significantly represented.

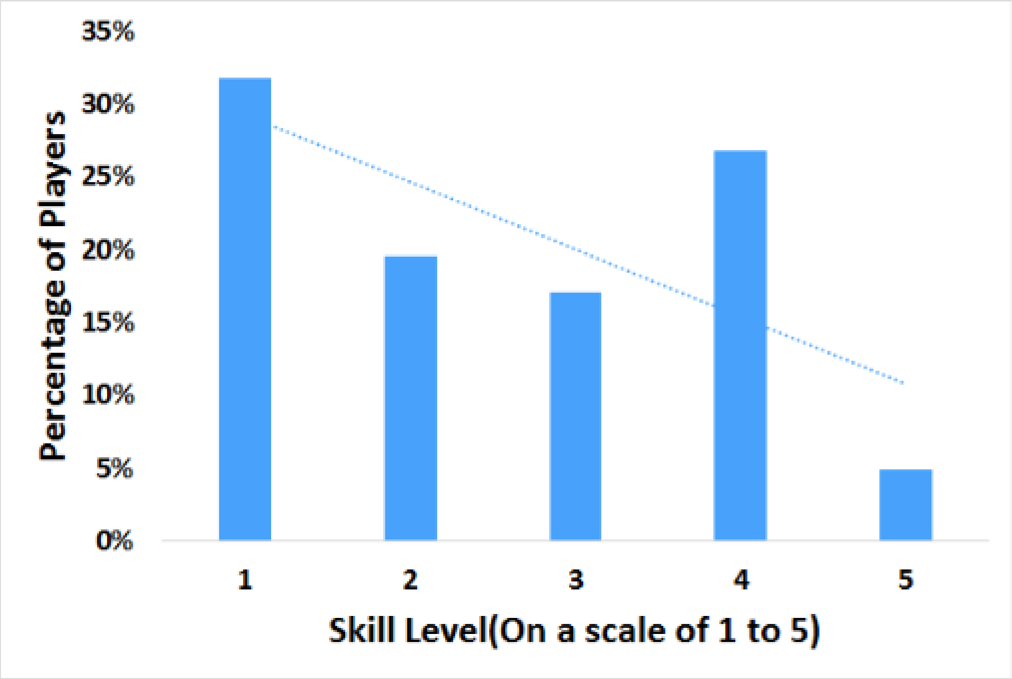


Figure 5.1-3 Skill Levels of First Person Shooters for participants with a gaming console

Figure 5.1-3 shows how players have rated their skills with FPS console games on a scale of 1 to 5 going from lowest skill to highest skill. Most of our participants answered 1 out of 5, while the second largest number answered 4 out of 5 and the least number of people gave the answer 5.

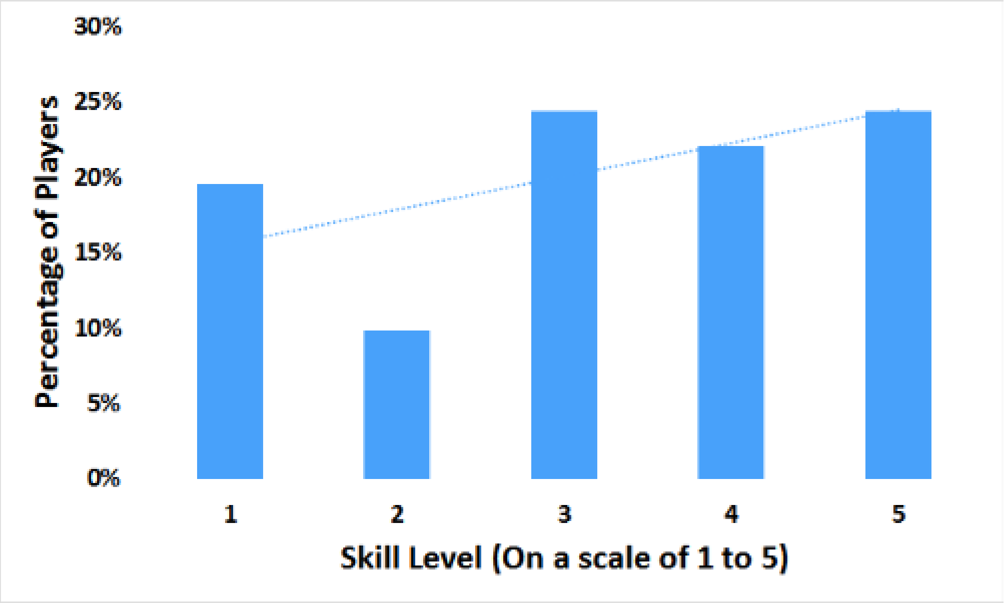


Figure 5.1-4 Skill levels of participants with a PC

The next question is similar to the last, except it asks about PC FPS games as opposed to console FPS games. The breakdown is in Figure 5.1-4. Unlike the last question, a large percentage answered that their skill level was at least 3 or above, with about half of participants saying their skill levels were at least 4 out of 5.

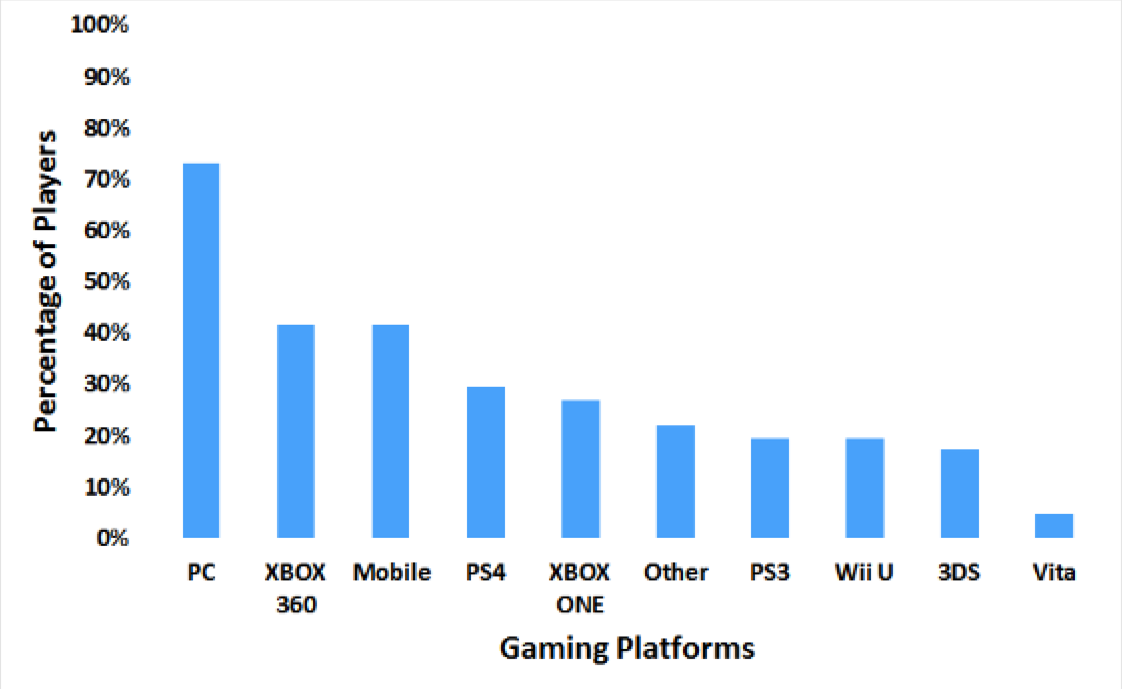


Figure 5.1-5 Recent gaming platforms used by participants

The breakdown to the next question "What platforms have you played games on in the last 6 months?" can be seen in Figure 5.1-5. A large majority answered PC while the next largest groups were the XBOX 360 and Mobile.

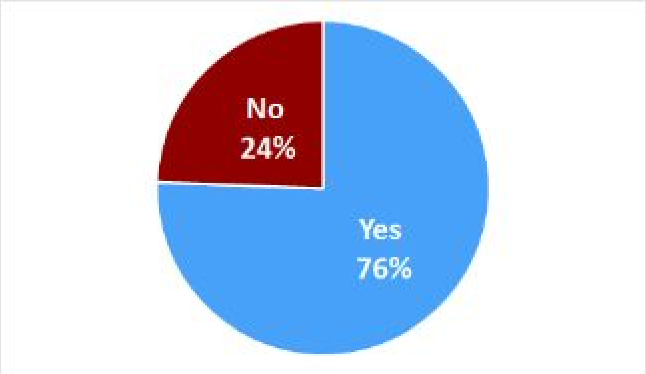
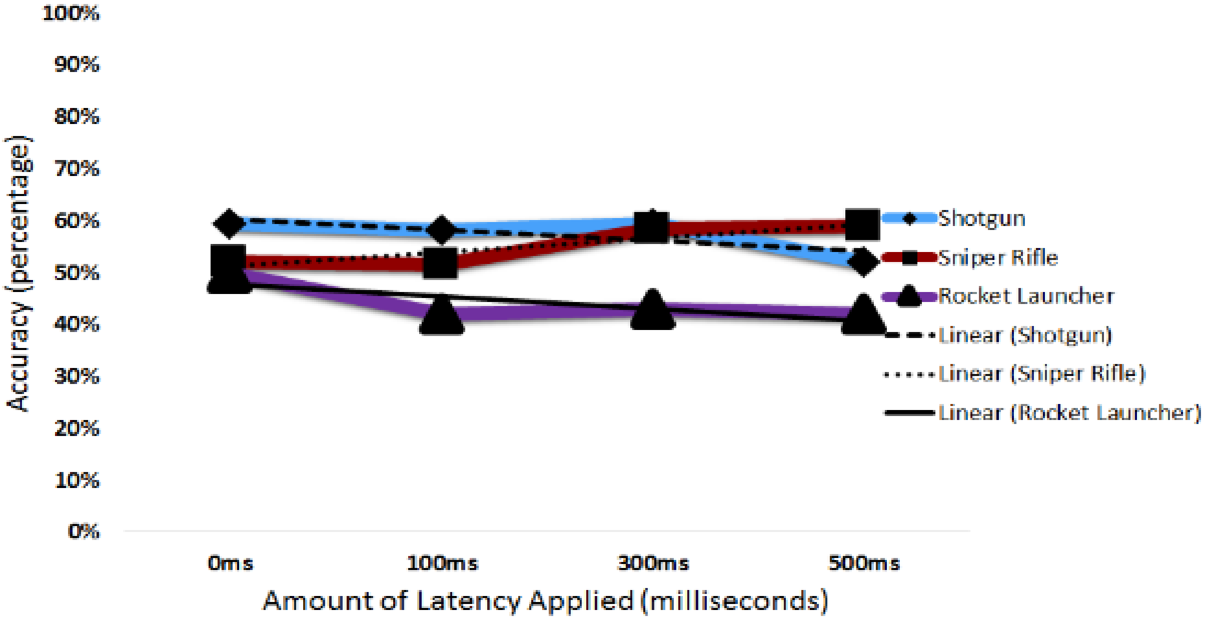


Figure 5.1-6 Participants that have played Team Fortress 2 in the past

The results to the final question in the Pre-survey, “Have you played Team Fortress 2 before?”, are depicted in Figure 5.1-6. About three quarters of the participants answered that they had played Team Fortress 2 before.

5.2 User Study Results

Below are the results that were collected from the user study that was explained in Section 4.4. We have broken down the data by calculating the average accuracy with latency, average win rate and the average damage between weapons.



	0 ms	100 ms	300 ms	`500 ms
Shotgun	59.36%	58.21%	59.40%	51.96%
Sniper Rifle	52.09%	51.58%	58.52%	58.85%
Rocket Launcher	49.94%	47.01%	47.89%	42.00%

Figure 5.2-1 Average accuracy of participants overall

SUMMARY				
Groups	Count	Sum	Average	Variance
Shotgun	4	2.289	0.572	0.00126
Sniper Rifle	4	2.210	0.552	0.00157
Rocket Launcher	4	1.768	0.442	0.00147

Figure 5.2-2 Summary of data for average accuracy of participants

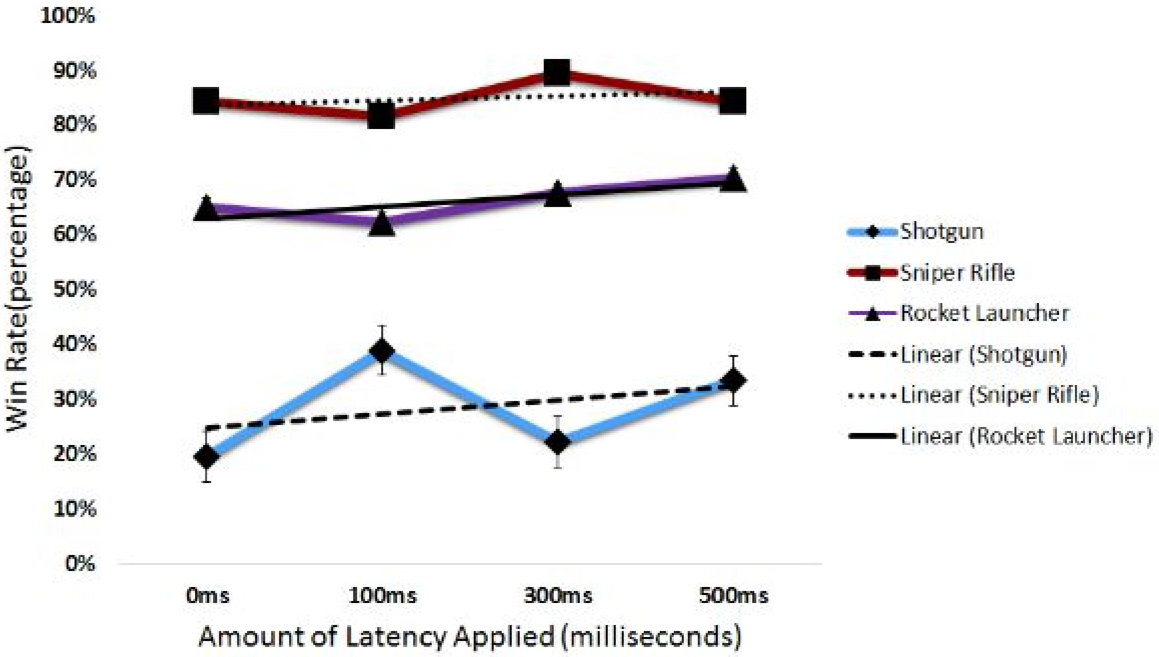
	R^2	Linear Correlation
Shotgun	0.646062	-0.80378
Sniper Rifle	0.830508	-0.911322
Rocket Launcher	0.45081	-0.671424

Figure 5.2-3 R squared value and linear correlation of data for average accuracy vs network latency of participants

ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.039411	2	0.019705	13.706	0.0018552	4.256
Within Groups	0.012939	9	0.0014376			
Total	0.052349	11				

Figure 5.2-4 Table showing statistical significance between weapons

Figure 5.2-1 represents the overall accuracy with each weapon across the different levels of network latency. Both the Shotgun and the Rocket Launcher accuracy decreased by about 7%, but the Sniper Rifle actually increased in accuracy. A simple linear regression is calculated to predict the accuracy of each weapon based on the amount of network latency applied. A significant regression equation was found between all weapons (F(2, 9) = 13.707, p < 0.002). We found that this is statistically significant because the p value is significantly less than the alpha value of 0.05. Correlations between each individual weapon and latency are located in Figure 5.2-3.



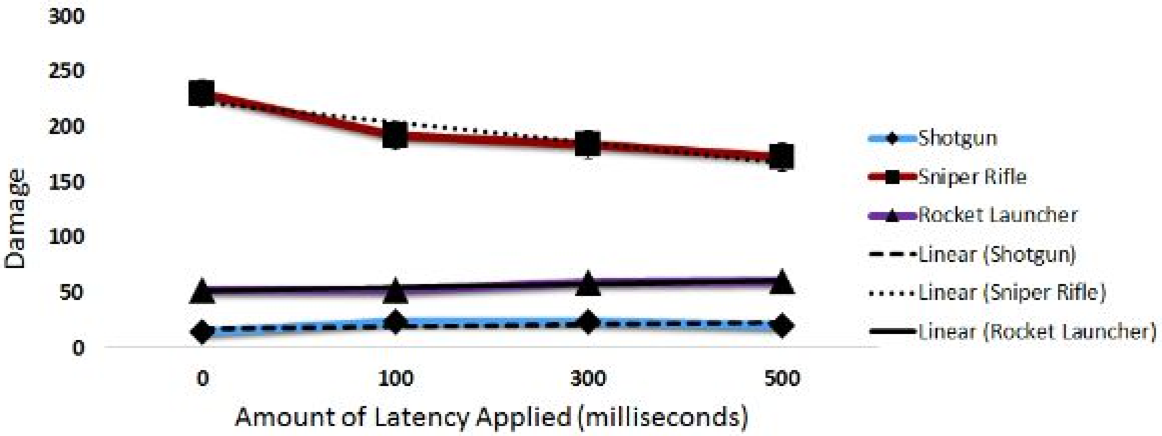
	0 ms	100 ms	300 ms	500 ms
Shotgun	19%	39%	22%	33%
Sniper Rifle	84%	82%	89%	84%
Rocket Launcher	65%	62%	68%	70%

Figure 5.2-5 Average win rate of all participants

	R^2	Linear Correlation
Shotgun	0.064007	-0.252996
Sniper Rifle	0.099002	-0.314646
Rocket Launcher	0.71959	-0.848286

Figure 5.2-6 R squared value and linear correlation of data for average win rate vs. network latency of all participants

Figure 5.2-5, depicts the average win rate with each weapon across the different levels of latency. For the most part, each weapon's win rate increases as the latency is applied.



	0 ms	100 ms	300 ms	500 ms
Shotgun	14	23	23	20
Sniper Rifle	230	192	184	172
Rocket Launcher	52	52	59	60

Figure 5.2-7 Average damage dealt by all players

SUMMARY				
Groups	Count	Sum	Average	Variance
Shotgun	4	80	20	18
Sniper Rifle	4	778	194.5	627.666
Rocket Launcher	4	223	55.75	18.916

Figure 5.2-8 Average damage of participants

	R^2	Linear Correlation
Shotgun	0.180791	-0.425195
Sniper Rifle	0.772739	-0.879056
Rocket Launcher	0.887105	-0.941863

Figure 5.2-9 R squared value and linear correlation of data for average damage vs network latency of participants

ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	67973.166	2	33986.583	153.419	1.1134E-07	4.256
Within Groups	1993.75	9	221.527			
Total	69966.916	11				

Figure 5.2-10 Table showing statistical significance between weapons

Figure 5.2-7 shows the average damage dealt each round, for each weapon and for each amount of network latency applied. A simple linear regression is calculated to predict the damage of each weapon based on the amount of network latency applied. A significant regression equation was found between all weapons, (F(2, 9) = 153.419, p < 1.113E-07). We found that this is statistically significant because the p value is significantly less than the alpha value of 0.05. Correlations between each individual weapon and latency are located in Figure 5.2-9.

5.3 Post Survey Results

The following sections contain the post-survey results for each weapon. Overall, our testers thought the network latency made the game more difficult, it was noticeable, it was not enjoyable, and it changed how they played the game. They were divided with how playable the game was, however, with a quarter saying 2 out of 7 and another quarter saying 5 out of 7.

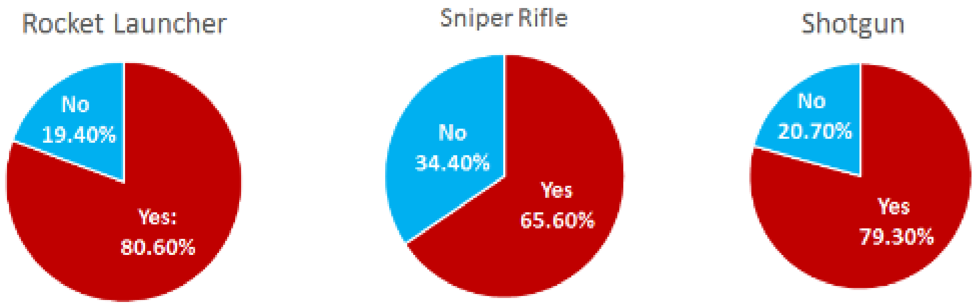


Figure 5.3-1: Amount of players that said latency affected the difficulty of the game

As expected, 5.3-1 shows that a large percentage of our testers felt like the added network latency made the game more difficult. This said, most of the participants found that the Sniper Rifle was affected less.

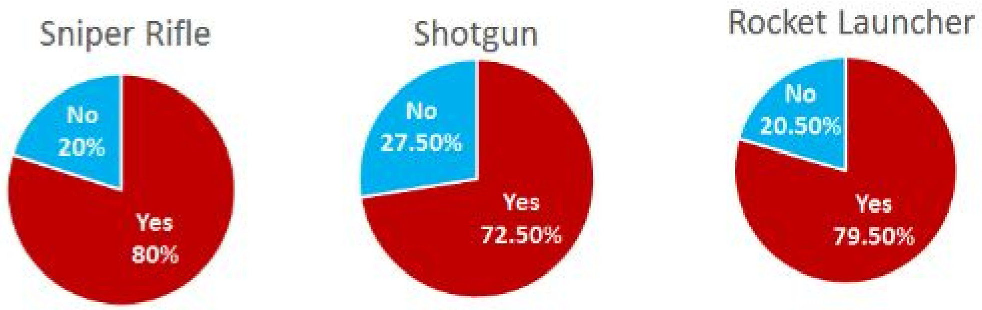


Figure 5.3-2 Amount of players that noticed the network latency

Figure 5.3-2 contains the breakdown for “Was the latency noticeable?”. Most of the participants noticed the artificial network latency, but fewer noticed it with the Shotgun.

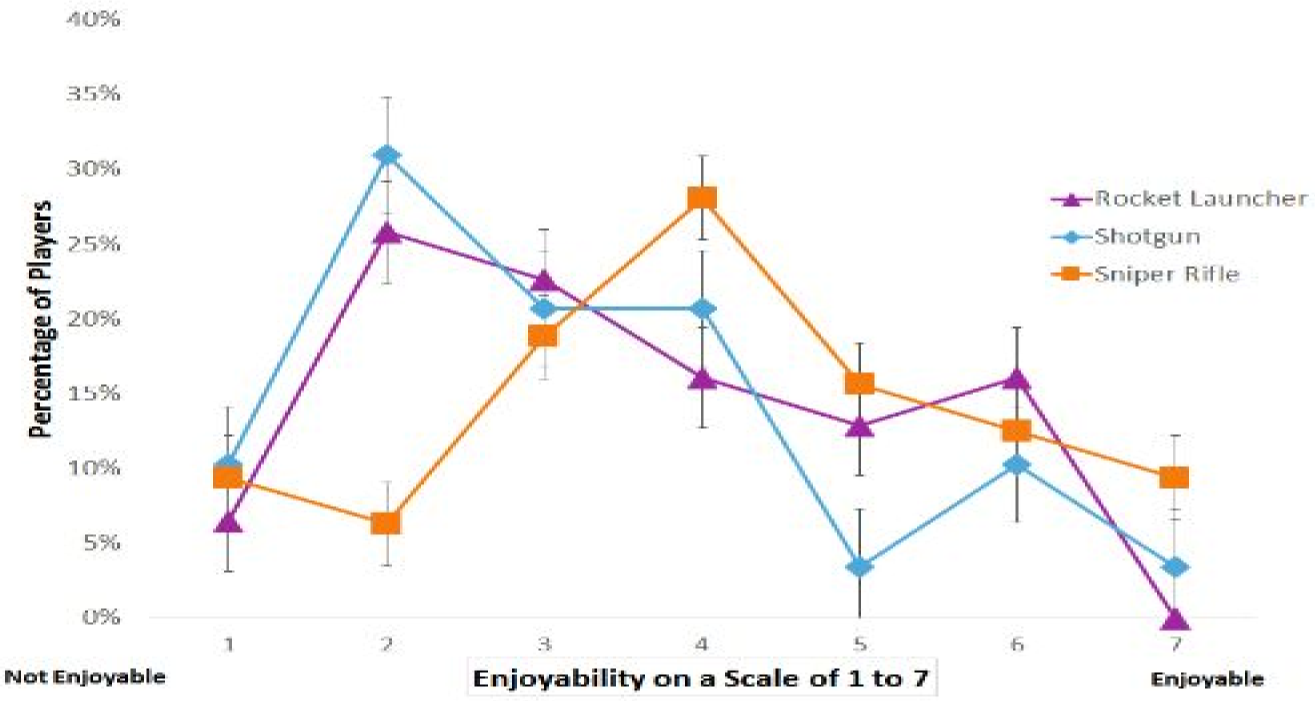


Figure 5.3-3 Reported enjoyability of game with network latency added

According to Figure 5.3-3, overall the players did not find the game very enjoyable with network latency added. The weapon that was the most fun to use with network latency was the Sniper Rifle.

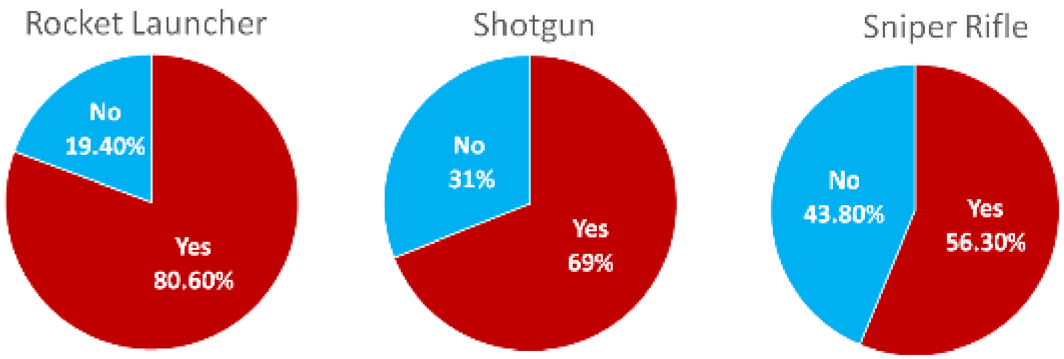


Figure 5.3-4 Amount of payers that changed how they played because of the network latency

Figure 5.3-4 has the breakdown for whether or not the players thought the added network latency changed how they played the game. Most replied that it had. Our participants changed their strategy the most for the Rocket Launcher and changed it the least for the Sniper Rifle.

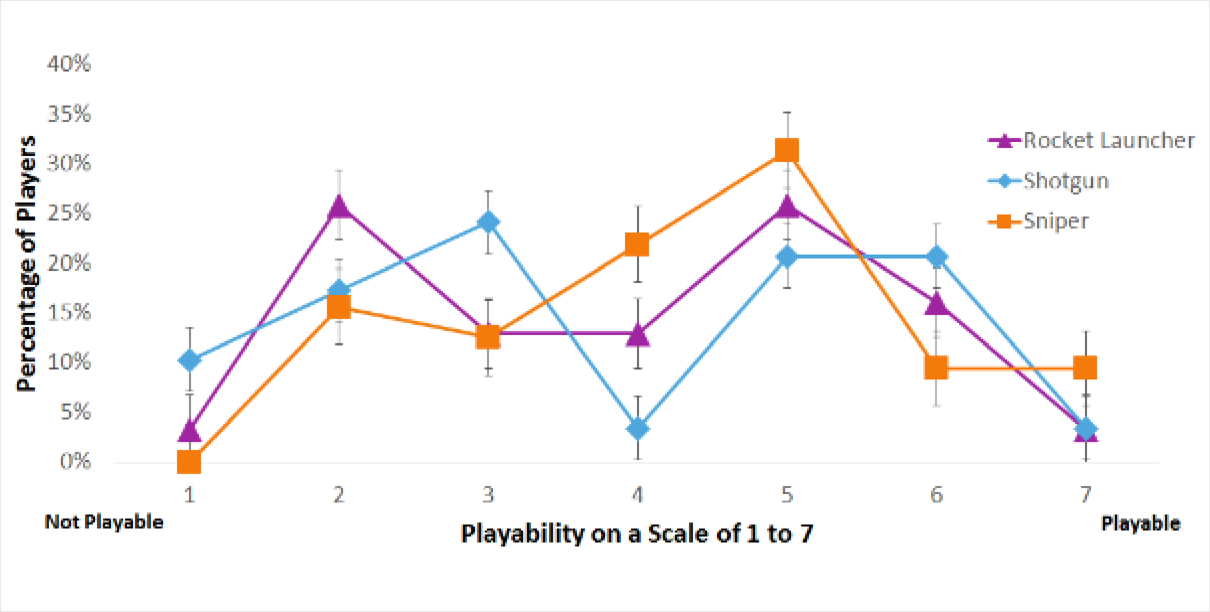


Figure 5.3-5 Reported playability of the game with network latency added

Figure 5.3-5 shows the results for the playability of the game with network latency added. For the most part our testers were fairly spread out with their replies. Our testers thought the Sniper Rifle was the most playable, even with the network latency.

6. Analysis / Discussion

This section analyzes the aforementioned data and draws conclusions on what we found.

6.1 Pre Survey Analysis

From the data that was collected from the user study, the majority of the participants had previous experience playing computer games. More specifically, most participants had more experience playing FPS games. More than three quarters of the participant pool has also played Team Fortress 2, but it is unknown if each of the players had prior experience with the amount of network latency that was tested in the user study.

6.2 User Study Analysis

From our user study, we have collected data that can be applied to the taxonomy that is being validated. First we compare the average accuracy, the average win rate, and then the average damage with the data from the previous taxonomy.

Overall, the least accurate weapon was the Rocket Launcher. Compared to the previous taxonomy, there was no measurement for the amount of accuracy. Since there is no area of effect based measurement in Team Fortress 2, our precision axes is determined by the overall accuracy of the player. The premise from the original taxonomy shows that with increased area of effect damage, there is increased expected impact regardless of the amount of latency that is applied.

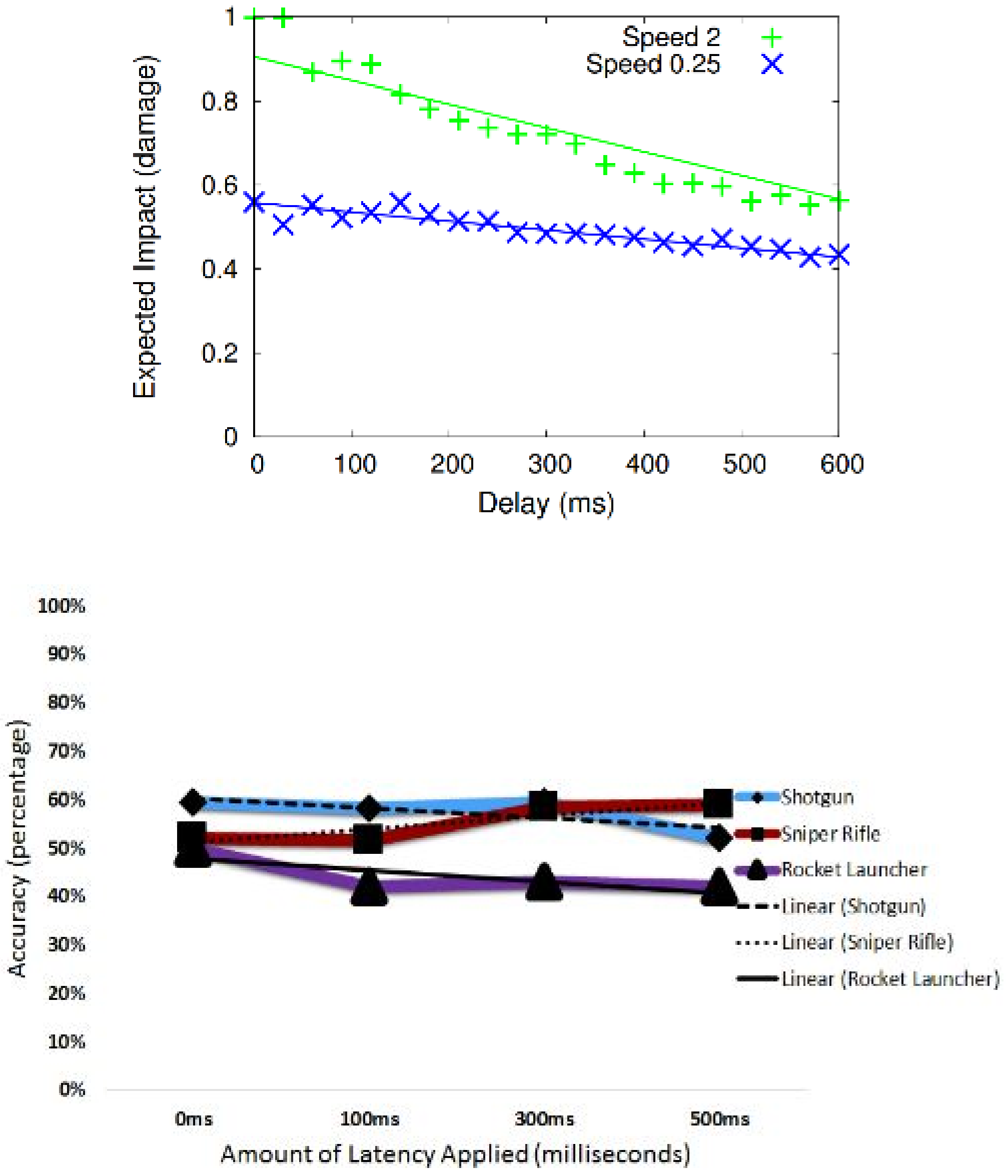


__Figure 6.2-1 Average accuracy of participants versus latency compared to the speed versus latency from the original taxonomy__

A notable comparison between our collected data and the previous taxonomy is the trend in the accuracy of the Rocket Launcher compared to the accuracy of the Shotgun. The data from 0ms network latency compared to the data from 500ms network latency with both the Rocket Launcher and the Shotgun decreases by about 7%. However, if you take the middle two network latency figures into account, it becomes clear that the Rocket Launcher has the greater expected impact (eg. amount of damage done) than the Shotgun. This contradicts the taxonomy because of two reasons. Firstly, according to the original taxonomy, weapons with a higher speed are more sensitive to delay. The Rocket Launcher uses a projectile which has a speed. The Shotgun, on the other hand, is faster than the Rocket Launcher because it uses Hitscan so the shot happens almost instantaneously. Also, the Rocket Launcher has a relatively large AoE, while the Shotgun fires multiple pellets in a single direction. According to the taxonomy, because of the Rocket Launcher's low projectile speed and larger AoE, it should be affected by the network latency less than the Shotgun. One reason why we think the Shotgun was affected less than the Rocket Launcher is because when testing for accuracy, the Shotgun will almost always hit the enemy when shot in their direction because it fires multiple pellets in a cone towards the enemy. Thus, only one of these pellets has to hit the enemy in order to trigger a shot hit event which has a fairly high probability of happening. Another way of thinking about it is that the Rocket Launcher has a larger AoE, but it's more difficult to place and aim than the Shotgun. With this in mind, and knowing that the Shotgun is a low-precision weapon, these trends make sense.

Unexpectedly, the accuracy of the Sniper Rifle increased as we increased the amount of network latency. From watching the participants, we believe that this is because as they played they found the dominant strategy for the Sniper which is to slowly peek a corner until the player can just barely see the bot around the corner. From this position, the user can see the bot but the bot can't see the player and you can easily line up a head-shot for a one-shot kill. Another possible reason for the increased accuracy is that some of the participants said that instead of going for head-shots, the user started going for easier body-shots as network latency was applied. Body-shots are easier to hit than head-shots, so the player would have an increased accuracy when using this strategy.

Moving on to the win rate, it seems that most of the graphs are not statistically significant with no real trends. We believe that this is mainly caused by random factors in the game that might change the outcome of a round eg. random critical hits and how accurate the bot is.

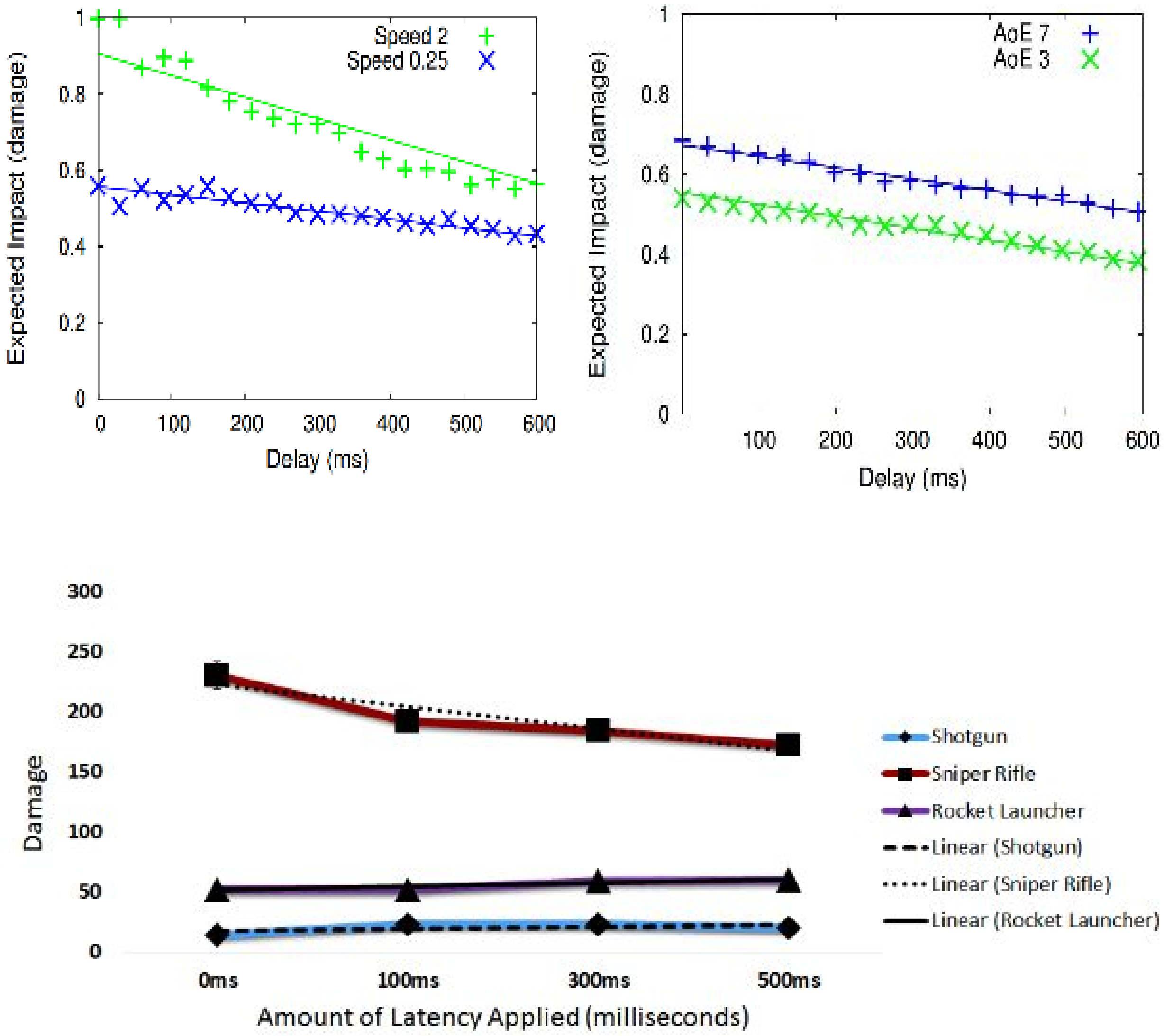


__Figure 6.2-2 Average damage dealt by all players versus latency compared to expected impact versus latency from previous taxonomy__

The damage dealt is similar to the win rate, in that the data for the Shotgun and the Rocket Launcher are not statistically significant. The Sniper Rifle, however, has a constant downwards trend as network latency is applied as we expected. We believe this is caused by the Rocket Launcher and Shotgun giving inconsistent damage per shot, while the Sniper Rifle's damage is always constant. The damage the player deals with the Shotgun is determined by the number of pellets from the Shotgun that hit the enemy, as well as the distance between the player and the enemy. The Rocket Launcher deals damage based on the distance between the enemy and the explosion of the rocket when it hits something. When scoped, the Sniper Rifle charges for 3.3 seconds, and increases in damage during this time. When its charge reaches the maximum, it will deal 150 damage for a body shot and 450 damage for a headshot. Note the Sniper Rifle can not get random critical hits because it will get a critical hit if the player gets a head-shot. Because of the afore-mentioned reasons, the damage that the player deals with the Sniper Rifle is more based on the player's performance instead of random events. We believe this to be the reason for the Sniper Rifle's downward trend in the damage graph.

6.3 Post Survey Analysis

Feedback from the participants indicates that the increase in network latency has made the aiming ability of the player much harder than with no added network latency. Some participants have said the following; "*Harder to aim and had to account for lag*" and "*I couldn't aim at all.*" These quotes are responses to the question "Did Latency affect how you would normally play the game?" As expected, the participants noticed that the added network latency had made them perform worse.

There are differences in responses about how players thought about conducting their next action or how they have thought about changing their strategy for playing the game. Some representative responses are; "*I had to pre-fire the rockets so that by the time I was around the corner they would leave the launcher,* " "*There was a delay from animations to actual shot that* *made it so that you had to lead even more than before. [D]ue to rocket launchers slow projectile you had to lead by a large amount,* ”and “*All I noticed was that the enemy reacted audibly later after being shot than normal when I was using the sniper rifle. Since this was when it was stationary, it's hard to judge but if it was moving more I would have had to adjust where I was shooting in order to hit it.*” From these responses, we can see that the players tried to change how they played the game to varying degrees of success.

According to the participants, the network latency was most noticeable with the Sniper Rifle, with the Rocket Launcher following closely behind. They found the network latency least noticeable with the Shotgun. When it came to the playability of the game, the participants said the Shotgun and Sniper Rifle were more playable, and most said the Rocket Launcher was not very playable. When asked about the difficulty that the network latency added, the participants said that the Sniper Rifle was least affected and then the Shotgun with the Rocket Launcher being the most affected by the network latency. The order is the same for whether or not the network latency forced the players to adjust their playstyle. This order contradicts the taxonomy, since the Sniper Rifle is the most precise, and then the Shotgun, and then the Rocket Launcher. These are only the answers the participants gave us however. Overall, the results from the survey that was given after the study may prove to be a valuable asset in the creation of future studies.

7. Conclusion

As more games rely on online connectivity as a core feature, being able to classify and subsequently deal with latency is clearly desirable for both developers and researchers alike. In this study we attempted to validate a previous taxonomy to classify the impact of player actions while experiencing latency in networked video games. Our results indicated that lower precision weapons like the rocket launchers and shotguns were significantly less accurate when latency was increased. Additionally, the data showed that high precision weapons such as sniper rifles dealt less damage as latency increased. Ultimately, we were able to partially confirm the validity of the previous taxonomy. More specifically, our findings fit the taxonomy in regards to the impact of damage and the weapon's shooting speed on a player's performance but the results are inconclusive and require further research on other aspects of player actions. Our results will most likely be utilized to aid future research on validating the taxonomy.

7.1 Future Work

We were limited by the number of participants that were willing to sign up for our study, and as a result we ended up having a large variance in skill level. This became problematic when participants did not have any familiarity with FPS games. While the practice period gave them some time to get familiar with the basics, they still exhibited a different type of playstyle compared to experienced participants. When considering the number of ways that results could be skewed by individual gameplay variance between trials, a much larger number of similarly skilled participants would be able to improve future studies tremendously.

Another problem we encountered with the study was the poor bot pathfinding due to the game mode that was used for the study, Arena Mode. Basically, they were not able to move outside of a small and confined area of the map. While this was good for repeatability, it severely hampered the dynamics of gameplay and caused rounds to become repetitive and the tactics required to defeat the bot became trivial. Perhaps most importantly, the poor navigation of the bot made the effects of network latency practically non-existent in some scenarios. For future trials, it is important to have consistent yet dynamic enemies that actually move about the map and fully expose the issues inherent in network latency when two moving players are involved.

We also had issues with the network latency simulation not fully replicating all of the effects of network latency. While we attempted to turn off all forms of latency compensation, some of the effects of network latency could not be simulated with our toolset. Future trials should look into software, either custom or off-the-shelf, that better simulates all the aspects of network latency while preventing compensation measures from unintentionally engaging.

One decision we made was to keep the order of network latency escalation consistent across all trials. We conducted the tests starting with 0ms of network latency added, and then increased it each round. While this made data collection significantly easier, it compromised our ability to distinguish the effects of network latency from the effects of participants naturally gaining experience by playing the game. Further research should be considered where the order of the network latency alterations is randomized across participants.

8. Bibliography

[1] Claypool, Mark, Tianhe Wang, and Mcintyre Watts. "A Taxonomy for Player Actions with Latency in Network Games." Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video - NOSSDAV '15 (2015). Web.

[2] Beigbeder, Tom, Rory Coughlan, Corey Lusher, John Plunkett, Emmanuel Agu, and Mark Claypool. "The Effects of Loss and Latency on User Performance in Unreal Tournament 2003®." Proceedings of ACM SIGCOMM Workshops on NetGames '04 Network and System Support for Games - (2004). Web.

[3] Claypool, Mark, and Kajal Claypool. "Latency Can Kill." Proceedings of the First Annual ACM SIGMM Conference on Multimedia Systems - MMSys (2010). Web.

[4] Y. Ishibashi and H. Kaneoka, "Fairness Among Game Players in Networked Haptic Environments Influence of Network Latency," *Multimedia and Expo,. ICME. IEEE International Conference*, pp. 4 pp.-. (2005)

[5] Howard, Eben, Clint Cooper, Mike P. Wittie, Steven Swinford, and Qing Yang. "Cascading Impact of Lag on Quality of Experience in Cooperative Multiplayer Games." 13th Annual Workshop on Network and Systems Support for Games (2014). Web.

[6] Henderson, Tristan. "Latency and User Behaviour on a Multiplayer Game Server." *Networked Group Communication*. Springer Berlin Heidelberg (2001)

[7] Nathan Sheldon, Eric Girard, Seth Borg, Mark Claypool, Emmanuel Agu. "The Effect of Latency on User Performance in Warcraft III." Proceedings of the 2nd workshop on Network and system support for games. ACM (2003)

[8] Jeremy Brun, Farzad Safaei, and Paul Boustead. “Managing Latency and Fairness in Networked Games”. Commun. ACM 49, 11 (2006), 46-51.

[9] Claypool, Mark. "The Effect of Latency on User Performance in Real-Time strategy games." *Computer Networks* 49.1 (2005): 52-70.

[10] L. Pantel and L. C. Wolf. “On the Impact of Delay on Real-Time Multiplayer Games”. Proceedings of the 22nd ACM Workshop on Network and Operating Systems Support for Digital Audio and Video - NOSSDAV '12, (2002)

[11] T. Fritsch, H. Ritter, and J. H. Schiller. “The Effect of Latency and Network Limitations on MMORPGs: a Field Study of Everquest 2”. In NetGames, Hawthorne, NY, USA, (2005)

[12] Armitage, Grenville. "An Experimental Estimation of Latency Sensitivity in Multiplayer Quake 3." Networks ICON. The 11th IEEE International Conference IEEE (2003)

[13] K.-T. Chen, Y.-C. Chang, H.-J. Hsu, D.-Y. Chen, C.-Y. Huang, and C.-H. Hsu. “On the Quality of Service of Cloud Gaming Systems”. IEEE Transactions on Multimedia, 26(2), (2014)

[14] R. Shea, J. Liu, E. Ngai, and Y. Cui. Cloud Gaming. “Architecture and Performance”. IEEE Network, 27(4) (2013)

[15] M. Claypool and D. Finkel. “The Effects of Latency on Player Performance in Cloud-Based Games”. In NetGames, Nagoya, Japan (2014)

[16] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoβfeld. “An Evaluation of QoE in Cloud Gaming Based on Subjective Tests”. In IEEE Innovative Mobile and Internet Services in Ubiquitous Computing, pages 330–335, Seoul, Korea (2011)

[17] M. Claypool and K. Claypool. “Latency and Player Actions in Online Games”. Comm. of the ACM (2006)

[18] "MedicStats SourceMod Plugin." Web. (2016). SupStats2 Plugin Description and Download
http://www.teamfortress.tv/13598/medicstats-sourcemod-plugin/

[19] "SourceMod: Half-Life 2 Scripting." Web. (2016)
http://www.sourcemod.net/

[20] "[TF2] Weapon Limiter - AlliedModders." Web. (2016)
https://forums.alliedmods.net/showthread.php?p=1761871/