A Study on the effects of Network Latency on Peeker's Advantage in FPS Games

An Interactive Qualifying Project Report
Submitted to the faculty of the
WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the Degree of Bachelor of Science
By

Colin Mettler
Dexuan Tang
Zesheng Chen

Submitted to
Professor Mark Claypool
Worcester Polytechnic Institute

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

March 2023

Abstract

Peeker's advantage - the performance advantage that the player peeking around the corner has over the player defending the corner in FPS games - is commonly believed to be caused by network latency. However, latency's impact on peeker's advantage has not been extensively studied. This paper examines the effect of latency and peeker's distance from the corner on peeker's advantage through a 24-person user study conducted using an open-source FPS game: First Person Science. One of three pre-set latencies and distances were assigned to the players for each game randomly. The analysis of the hits, wins, accuracy, and time-to-damage data shows that peeker's advantage is more affected by the defender's latency than the peeker's latency, and the optimal distance for peeking does not vary with latency.

Contents

1 Introduction

FPS games and E-sport events have become popular forms of entertainment and a proliferating industry. The FPS game market in the US is projected to become a multi-million dollar industry by 2029. In the US, the E-sport industry achieved 282.16 million dollars in revenue in 2022 [1]. This is why we seek to improve the user experience and fairness in FPS games as the market grows.

Understanding the impact of latency on user performance is an important aspect of making the user experience better. For many FPS games, such as Counter-Strike: Global Offensive (CS:GO) (released in 2012), or Valorant (released in 2020), there is a concept referred to as peeker's advantage. Peeker's advantage is an artifact of networked gameplay that refers to an advantage that someone peeking around a corner has over an opponent on the other side, typically believed to be caused by network latency from the game clients to the server.

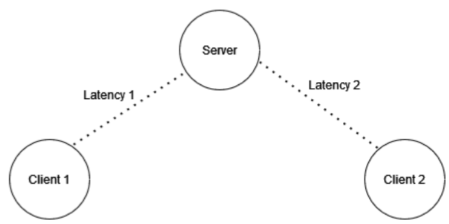


Figure 1: Diagram of a simple server and client setup.

Consider a simple server-client architecture shown in Figure 1. Each client experiences a network latency when data exchange occurs between the server and the client, since the client has to send the player's commands to the server and execute the server's response. It takes half of the player's network round-trip for those commands to reach the server, at which point the server executes them on the server's authoritative copy. The server then returns the result to the client, which takes the second half of the player's network round-trip.

Peeker's advantage caused by network latency is explained in Figure 2. In Figure 2a, at t=1, one player is holding a corner, the defender, and the other player is preparing to peek around that corner, the peeker. In Figure 2b, the peeker has moved around the corner at t=2 and can now see the defender, who has not moved. However, the defender's client lags behind the server and the peeker's client., and as a result, the peeker sees the defender first and can react first and shoot. With the result being Figure 2c, the defender will die at t=3 when the peeker has started moving back behind the corner for cover.

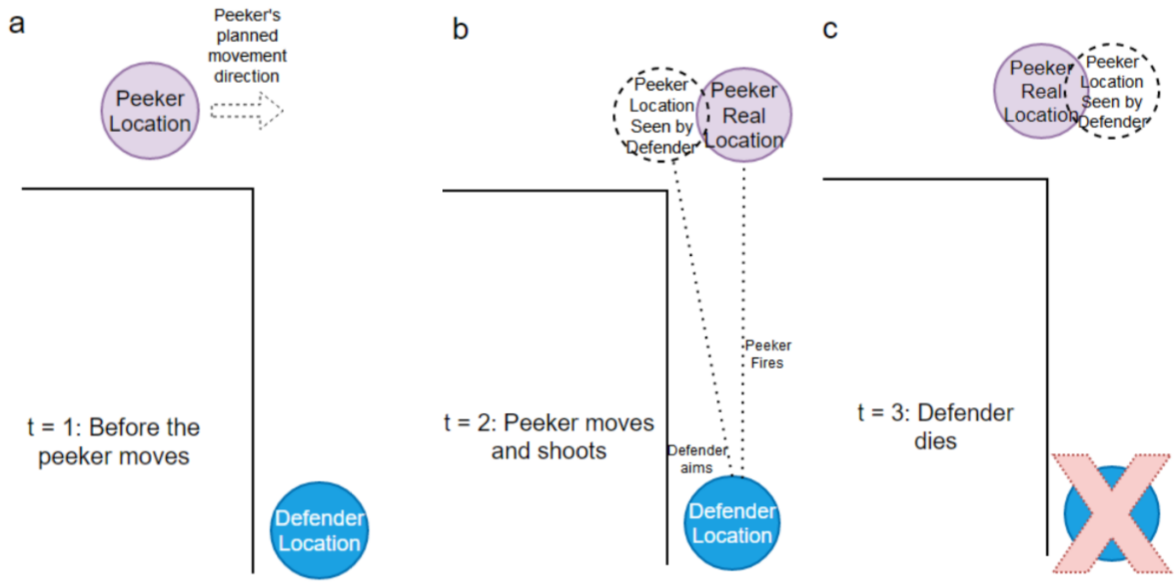


Figure 2: Overhead illustration of peeker's advantage.

Figure 3 illustrates the peeker's advantage in a timeline. There are a few frames of time where a player peeks around a corner first and spots their opponent (defender), and their opponent still has no idea that they exist. Peeker's advantage applies to most commercial FPS shooter games, including Valorant and CS: GO. It can be eliminated by having players with the same minimal latency in a LAN environment but is unavailable for games played over the Internet. It is particularly important in professional competitions, where the reflex time of the players is so fast and skills are so high that the difference between losing or winning a gunfight can be a few milliseconds. Even a minor Peeker's Advantage could give a professional player a significant advantage in online conditions.

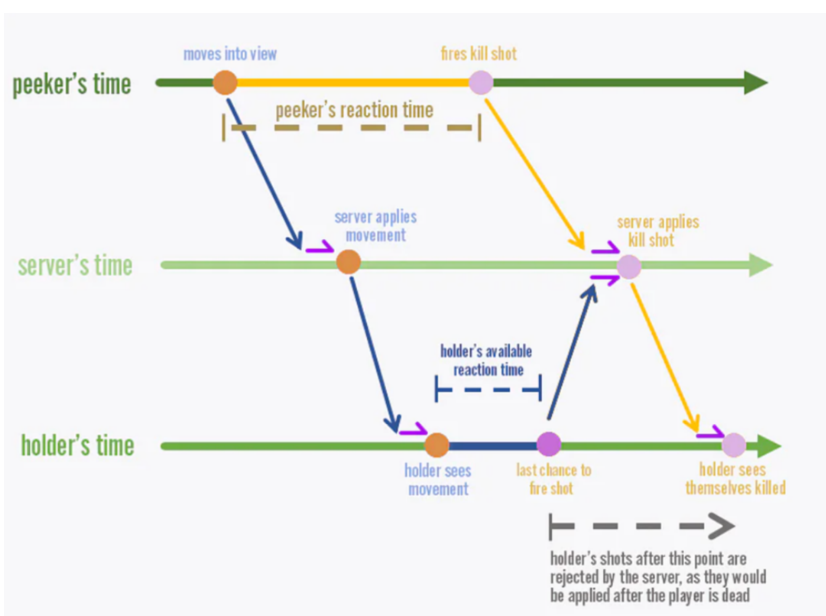
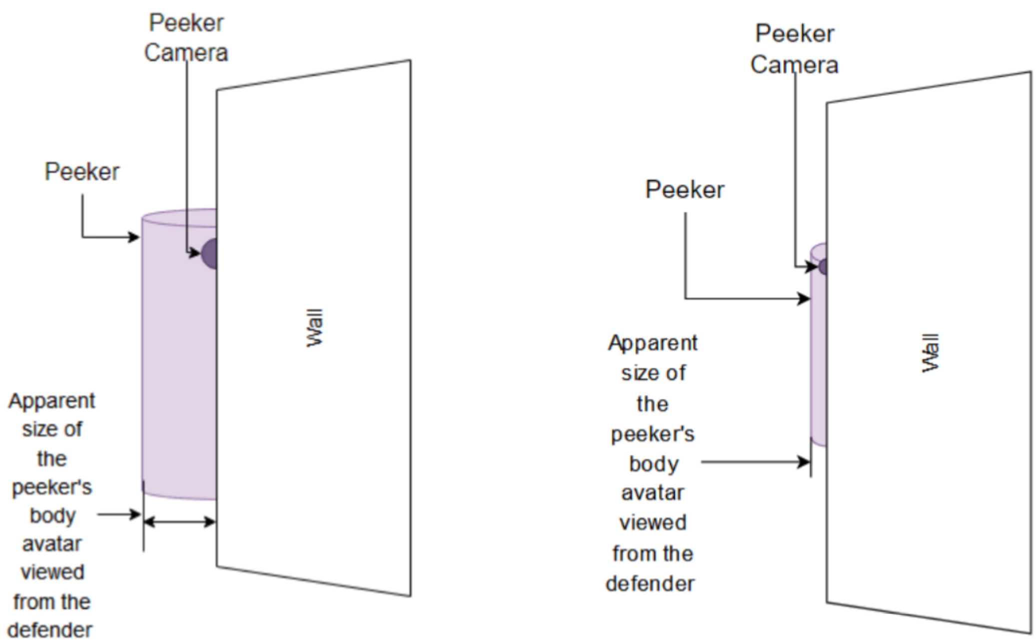


Figure 3: Timeline illustration of Peeker's Advantage [2]

Typically in FPS games, it is better to peek farther from the wall due to geometry. The closer the peeker is to the wall, the more of the peeker's avatar that can be seen before it can see around the corner. Figure 4 shows the peeker's avatar viewed by the defender when the peeker is close to the corner (Figure 4a) and far from the corner (Figure 4b). The peeker's camera is at the avatar's center in both cases. In Figure 4a, the apparent size of the peeker's avatar that is sticking out of the corner to allow the peeker to have a line of sight to the defender is bigger than the one in Figure 4b. However, if the distance between the two players is too large, the apparent size of the enemy avatar becomes too small to be easily hit. Thus, we assume that there is a best distance wherein the peeker is farther from the corner than the defender without being so far as to make the defender too small to target effectively.



(a) When peeker is close to the corner

(b) When peeker is far from the corner

Figure 4: Illustrations of peeker's avatar geometry when viewed by the defender in first person perspective.

Despite the prevalence and awareness of peeker's advantage among players, there are little to no studies measuring this effect. The lack of scientific research behind peeker's advantage also may have brought confusion regarding theories that peeker's advantage is not caused by network latency but rather pre-firing, the act of anticipating an enemy player before peeking to eliminate the need for reaction time, and crosshair placement, aiming at common angles before peeking to eliminate as much need to move the mouse as possible.

This study seeks to address the lack of analysis of peeker's advantage by studying peeking versus defending at various network latencies and distances to determine the extent to which latency gives the peeker an advantage. This study hypothesizes that:

1. The effectiveness of peeking versus defending is more affected by the defender's latency than by the peeker's.

2. There is an optimal distance between the peeker and the corner, which does not vary with latency.

To test these hypotheses, we conducted a user study of a peeking-defending scenario with pairs of users on a server-client structure using an open source FPS game. The game assigned different roles, latencies, and distances to the players randomly. We collected hits, time-to-damage, wins/losses, reflex time, and Quality of Experience answers for analysis.

Analysis of the result shows that the variation in the defender's latency has a greater effect on peeker's advantage. The advantage of the peeker in terms of hits and win rates increases as the defender's latency increases. The peeker's latency has a relatively small effect on the peeker's advantage.

Both the peeker and the defender have the highest number of hits when the distance of the peeker is closest to the corner. Peeker's advantage is the greatest when the peeker is closest to the corner regardless of latency.

The rest of this report is orgranize as follows: Chapter 2 lists related research on peeker's advantage. Chapter 3 provides details on the experiment's methodology. Chapter 4 reports on the data collected, and summrizes findings based on these data, Chapter 5 provides suggestions on possible future works, and Chapter 6 concludes this study.

2 Related Work

This section covers studies related to this one in several different areas. It is split up into sections on Network Latency, Lag Compensation, Accuracy, and Movement.

2.1 Network Latency

Many studies have been conducted on the effect of network latency in games. Players' performance in FPS games is very sensitive to latency changes because games in this genre are usually fast-paced and require the user to target accurately.

It is often believed that skill could offset network latency since higher-skilled players may be better at performing tasks in games subconsciously. A study by T. Henderson has studied the relationship between skill and latency [3]. The author set up Half-Life (1998) game servers at University College London and measured round-trip network latency. Most players have round-trip latencies of between 50 ms and 300 ms. The author calculated skill based on the player's kills, death, and weapon type. The author observed a negative correlation between skill and delay and concluded that a lower delay may lead to improved performance.

Real-world latency data and players' tolerability of latency in commercial games were studied by C. Jiang et al [4]. in a survey of over 240 users and 550 games. The survey included players' self-reported rating of their ability, preferred games, preferred devices, Internet connection, self-reported minimum and maximum network latency, and how often network latency affects their performance. In general, users reported that latencies as low as 85 ms were noticeable but about 600 ms latency can be tolerated. However, most users claimed that the maximum tolerable latency in FPS games is about 180 ms. Most users also rated FPS games as fast-paced. This study and T. Henderson's study helped the formulation of the latencies used in our study to be close to what player's experience in the real world.

S. Liu et al. conducted multiple studies on the effect of network latencies. The results from these study also provides baseline for forming our hypothesis and what should relationships should be expected in the result of this study. They measured player accuracy, score, and Quality of Experience in CS:GO games with 20 AI-controlled players under network round-trip latency ranging from 25 ms to 150 ms and with latency compensation on and off [**?**]. The accuracy decreased by 3% and the score decreased by around 17% as latency increased to 150 ms. Quality of Experience also show a significant difference of 25% between 25 ms and 150 ms. The same authors also compared how latency affects players with different skill levels differently using a similar user study. [5]. In this study, higher-skilled players are those who have experienced CS:GO and FPS players and lowered-skilled players are those who have not experienced CS:GO players but are skilled FPS players. The authors' results showed that latency affected the score, accuracy, and Quality of Experience of higher-skilled players more than lower-skilled players.

Local and network latencies in CS:GO were compared by S. Liu et al [6]. This study measured the effects of both local and network latency on accuracy of CS:GO players, score as points per minute (points/minute), and Quality of Experience (QoE) on a 5 point scale. Local latency was found to have a greater effect on players' performance than network latency. Specifically, a decrease of 100 ms of local latency increased accuracy by 6%, score by 3 points/minute, and QoE by 1.6/5 points, whereas a decrease of 100 ms of network latency only resulted in accuracy increasing by 2%, score by 2 points/minute, and QoE by 0.7/5 points.

2.2 Lag Compensation

Lag compensation is not something covered by the scope of this study, however it is related to network latency since lag compensation is an attempt to reduce the effects of network latency. FPS games use lag compensation due to being generally more sensitive to latency than other types of games. There are a variety of different ways to implement lag compensation. Shengmei Liu, Xiaokun Xu, et al studied the efficacy of many of these methods. [7]. The broad grouping of methods within the paper are Feedback, Prediction, Time Manipulation, and World Adjustment. Feedback, Time Manipulation and Prediction types of lag compensation techniques tend to be the most common within FPS games. Generally speaking Prediction works by estimating the game-state of a player based on world information, the game rules, and the previous known game-state. Time manipulation works by altering the in-game time for computing game-states and resolving player actions. The relevance of this study is because lag compensation is often cited as attempting to minimize peeker's advantage. Different lag compensation techniques do this in different ways and can perhaps have different effectiveness in minimizing peeker's advantage.

There are not many studies that focus directly on the effects of lag compensation on Peeker's Advantage, but one study by S. W. K. Lee and R. K. C. Chang focuses on something similar where a player finds themselves being killed after "unpeeking" - moving back behind - a corner [8]. This can occur in games such as "Counter-Strike Global Offensive" and is caused by lag compensation, usually Time Manipulation. However it can also be caused by network latency under certain conditions. This study focused on the effects of network latency and player movement speed on the "shot behind a corner" phenomenon when players are rolled back due to lag compensation, as well as the perceived fairness of these rollbacks. The study concluded that rollback distances of up to 0.6m and at latencies of up to 250 ms did not affect the fairness of the game. This study is relevant as it showcases the relationship between lag compensation and a scenario that is mechanically very similar to peeking and peeker's advantage.

2.3 Accuracy

There are numerous resources available that highlight the significance of accuracy in the field of first-person shooting (FPS) games. Accuracy is a crucial factor in such games, as players need to develop their accuracy skills in order to overcome adversaries and triumph over the game.

A multitude of factors can affect accuracy, such as a player's current physical and mental state. Fatigue and poor health can negatively influence accuracy, and therefore affect a player's performance. Additionally, accuracy is dependent on the rate at which frames are displayed in a computer game. Kajal T. Claypool et al noted that actions requiring precise, rapid response, such as shooting, are significantly affected by fluctuations in frame rates. [9]. This is relevant as we hoped to isolate the effects of network latency. In this case, it was using identical computers and monitors to ensure that the frame rates were as close as possible. In addition players used the same mice and keyboards to reduce variability.

However, an important factor affecting accuracy is network latency. Ragnhild Eg, Kjetil Raaen, et al found that even minimal delays can have a substantial impact on the player's overall gaming experience, potentially leading to reduced task performance and lower quality of experience. [10]. Kajal Claypool, Feissal Damaa, et al reiterates that frame rates and resolutions directly affect player performance, and ultimately impact the overall playability and enjoyability of the game, including accuracy. This is relevant as the effects of network latency on accuracy could also be a strong factor behind certain results from this study [11].

On the other side, as previously mentioned, the impact of server-side factors on accuracy cannot be disregarded, but the influence of client-side factors must also be considered. In the majority of competitive games, the players' ability to maintain accuracy is still the most important factor in winning the match, thus the client-side impact on accuracy remains significant for FPS games.

2.4 Movement

Movement is another essential part of FPS gameplay and our study. The players have to be able to navigate the map and position themselves effectively in order to target the opponents and avoid being shot. Two papers have studied the effect of latency on player movements.

S. Liu and M. Claypool conducted a 30-person user study using a custom hide-and-seek game to study the effect of latency on navigating a square room with obstacles. [12]. The authors used a similar server-client architecture to our study and tested both local latency and network latency. Both network latency and local latency shows similar effects: as latency increased, the player score decreased linearly, and the seeker sees the hider for a shorter period of time per round. The study concluded that latency has strong impacts on players' ability to score and position themselves. This has relevant connections to our study as positioning is very important to the strategy and technique aspect of peeking. A perfect peek requires the player to have good crosshair placement, to only be peeking one angle at a time, not over peeking angles, and more. If a player struggles to successfully position themselves, it can harm their ability to successfully peek in a real match.

X.Xu et al. analyzed the length of time of the player's movement per minute. The authors used a server and a single client with network round-trip latencies ranging from 0 ms to 100 ms. [13]. Twenty AI-controlled players were added to the game, with achieving as many kills as possible as the objective of the game. As latency increases, the player's avatar movement decreases. The authors suggested that this could be caused by the shortened survival time as the players have a harder time avoiding being shot and a harder time moving into position at higher latencies. The study noted that this effect is greater when the players use the shorter-ranged Nova shotgun. This study is very relevant to our own study. If a player's movement per minute changes as latency changes that also means that a player is far more likely to under or over peek, especially if they over or under compensate for the latency. Without the skill and ability to compensate properly this could very much affect the peeker's advantage.

3 Methodology

To investigate how network latency may affect the peeker's advantage in First-Person Shooter (FPS) games, we conducted the experiment in a way that added controlled amounts of network latency, recruited players for a user study, and measured player performance and quality of experience.

3.1 Game Description

We conducted the study in an on-campus computer lab in a 1v1 setting for 30 minutes per pair of players using a client-server architecture shown in Figure 1. We used a custom FPS game: First Person Science (FPSci). It is a tool developed by Nvidia for conducting user studies on FPS-style tasks. The WPI MQP team has added a client-server relationship to this tool, allowing for the use of a multiplayer set up with 2 clients and 1 server, wherein the clients can compete against each other. FPSci allows us to set independent variables including latency, spawn position, and move rates. FPSci is client authoritative. When at least one player is hit by the other, both players are respawned.

Figure 5: A screenshot of FPSci's user interface.

3.2 Experiment Setup

This section describes how the experiment was set up. This includes the computer setup, our independent and dependent variables, the experiment design, and the tournament design.

3.2.1 Computer Setup

Two clients and one server were set up, as shown in Figure 1 in the introduction chapter. The latency for each client was controlled independently through a configuration file in FPSci. The latency adjustment was implemented by the WPI MQP team through the Latent Network class [14]. This was done with a client-server relationship, wherein the client was authortiative. The network library ENet was chosen to be used. A packet-based structure was used meaning that the client and server communicated by sending packets of data to each other. Latency was implemented by putting these packets onto a heap and holding on to them for the pre-set latency time before sending. This is why the latency values used are one-way, as the client would hold the packet for that amount of time before sending it to the server, and the server would also wait that amount of time before sending anything back. Both client computers were running Windows 10 on Intel Core i7 8700 CPUs, Nvidia GTX 1080 GPUs, 64GB of system memory, Lenovo 240Hz monitors, and Logitech G502 mice set to 800 DPI with pointer acceleration disabled. Local latency was measured with FPSci and live stream open using a high-speed camera. A window in FPSci was opened on each client, and the resulting mean local latency for 10 data points on each client was 26.2 ms, with a standard deviation of 3.2 ms. In FPSci, jumping was not enabled as part of isolating the peeker's movement to being lateral, counter strafing was enabled in order to simulate the movement of players in CS:GO and Valorant as the player base for this study was taken from players of those games, the weapon's reload time was set to 250 ms so that the time between shots was greater than the highest RTT latency to prevent players from spamming multiple shots and having them registered as hit before the server returned the respawn command from the first hit, and the peeker's move rate was set to 7 for all rounds, this was the default movement speed in FPSci. The client's gameplay screens were live-streamed to Twitch both as a method of recording both player's POV as well as to add a sense of competition.

3.2.2 Independent Variables

Independent variables listed in Table 1 were set in the FPSci's configuration file and were randomly combined and assigned to the users one combination per round by FPSci. To cover all combinations, there were 54 rounds in total per game. Each round elapses 15 seconds, and both players are respawned once a hit is registered. There is one respawn per trial. The study designers determined the distances and latency variables through pilot studies.

- Roles: Peeker, Defender
- Latency (One-Way): 0 ms, 30 ms, 60 ms
- Peeker Distance: 7 m, 19 m, 27 m
- Defender Distance: 25 m

In Figure 6, the peeker's movement was limited by low boxes in a custom map. These boxes have equal widths to allow the same front-to-back movement for the peeker under all distance conditions. By spawning the peeker in different boxes, we effectively changed their distance to the wall.

The defender's location was varied randomly from round to round to prevent the peeker from learning the defender's location and prefire. The variation was small enough not to affect the dependent variables noticeably. This was implemented by changing the angle between the defender and the wall by 3 degrees in a random direction each time a trial ends while keeping the defender's distance from the wall constant.

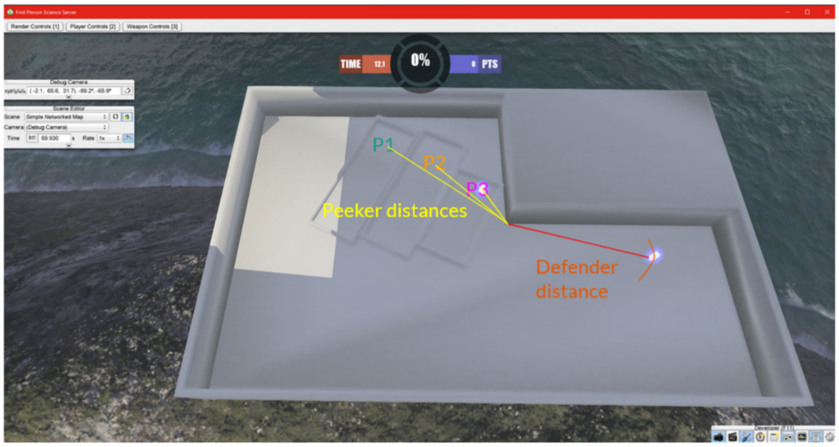


Figure 6: A screenshot of FPSci's user interface.

3.2.3 Dependent Variables

After each round, both players have given the following Quality of Experience Question that they could answer with their mouse pointer.

"Please rate the latency of the previous round"
"Options: 1(low) 2 3 4 5(high)"

FPSci generates log files for both the clients and the server. After each game, log files were collected and analyzed for the following dependent variables listed in Table 2. Mean reflex times were collected separately using a reflex time testing program [15] [16].

- Number of hits across all games for both players
- Wins/Losses
  - Rounds won by a player
  - Round win rates
- Time to Damage (TTD)
- Mean reflex time
- Quality of Experience (QoE) answers

3.3 Experiment Steps

1. Recruiting: A screener was sent to the potential users to measure their experience and skill levels in FPS games. The screener included questions about the player's most played FPS games, the self-reported number of hours and skills in each game, ranking, preferred mouse sensitivity, and whether or not they would like to participate in a tournament.

2. Informed consent: The users were given informed consent forms that informed them about the content of the game, live streaming, and covid protocols.

3. Reflex time testing: We tested the player's mean reflex time through the reflex time testing program. The users were asked to click the mouse as fast as possible when a square on the screen changed color. Mean reflex time across 10 trials was given by the program and recorded by the investigator.

4. Practice rounds: The practice rounds are identical to the ones used for recording data but without latencies and Quality of Experience questions. Users were given 18 practice rounds to get comfortable with the setup, as well as to ensure that their mouse sensitivities in a game match their preferred settings. Users were also allowed to adjust their mouse sensitivity during the practice rounds.

5. Experiment rounds: After players indicate that they are ready, the live stream and recording were started. FPSci executables used for experiment rounds were opened on both clients, and the users were asked to confirm their mouse sensitivity settings. The game started after both players were ready and FPSci looped through all variable combinations listed in Table 1. The user answered the Quality of Experience question after every round.

6. After the game ends, results were collected by copying the server and client log files that FPSci generated.

All personal information is held confidential. users' email addresses were collected only to contact them. Information will be deleted from our records within 60 days of the completion of the user study. Records of users' participation in this study will be confidential as the law permits. However, the study investigators, the sponsor, or its designee, and under certain circumstances, the Worcester Polytechnic Institute Institutional Review Board (WPI IRB) will be able to inspect and have access to confidential data. Any publication or presentation of the data will not identify the users.

For the user study, all users received a $10 Amazon eGift card as compensation for participating. They did not receive additional compensation for participating in the rematch. For the tournament, the winner received a graphics card, RTX 3090, provided by our sponsor, Nvidia, as the prize.

3.4 Tournaments

Furthermore, to evaluate peeker's advantage in more stressful, competitive games, we invited some players back to compete in further tests in a tournament-style competition for a graphics card sponsored by Nvidia. An example bracket is shown in Figure 7.

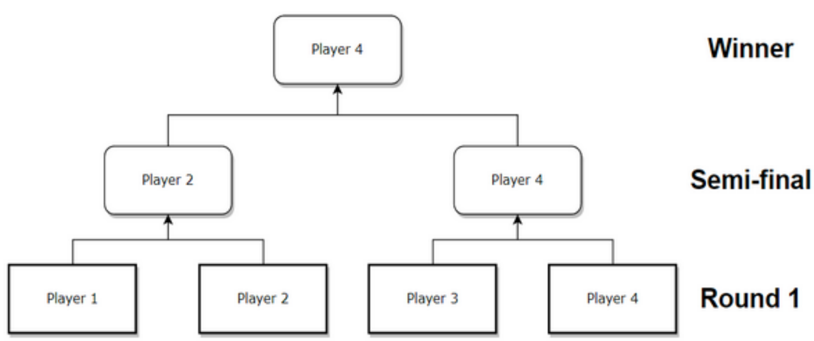


Figure 7: An Example of a single elimination tournament bracket.

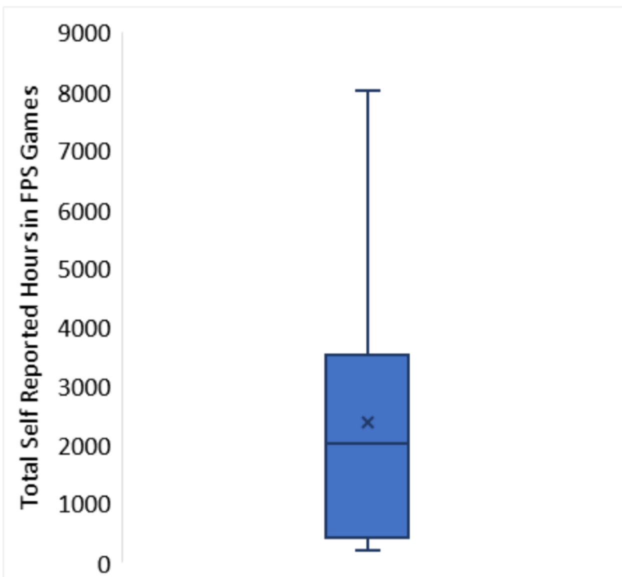
The tournament was a single-elimination tournament with a leaderboard style including semi-finals and grand final. In the single-elimination tournament, the players are matched based on skills, with the winner of a match being scheduled to play again against another winner, with the last player remaining at the end of the tournament winning. The higher round-win difference determines the winner. We also live-streamed tournaments, given the user's consent.

4 Analysis

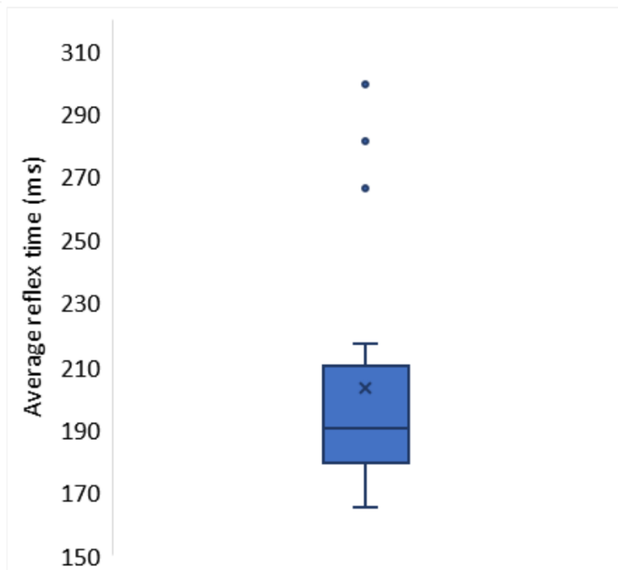
This chapter includes data analysis. Section 4.1 describes the details of the demographics of the users. Section 4.2 reports on the effect of latency combinations on player performance. Section 4.3 reports on the relationship between the peeker's distance to the wall, latencies, and player performances. Section 4.4 analyzes the time-to-damage data of both players, Section 4.5 reports on the relationship between reflex time and performance, Section 4.6 summarizes the Quality of Experience data, and Section 4.7 summarizes our analysis and compares the results to our hypothesis.

4.1 Demographics

24 WPI students participated in this study indicating some level of experience playing competitive FPS games on a PC with a keyboard and a mouse. The self-reported total hours in FPS games are graphed in Figure 8a. The mean reflex time for each player was recorded and compiled into the box and whiskers plot in Figure 8b. The "x" in the box plots denotes the mean. Local latency of 26.2 ms was not subtracted from reflex time data.



(a) FPS Hours



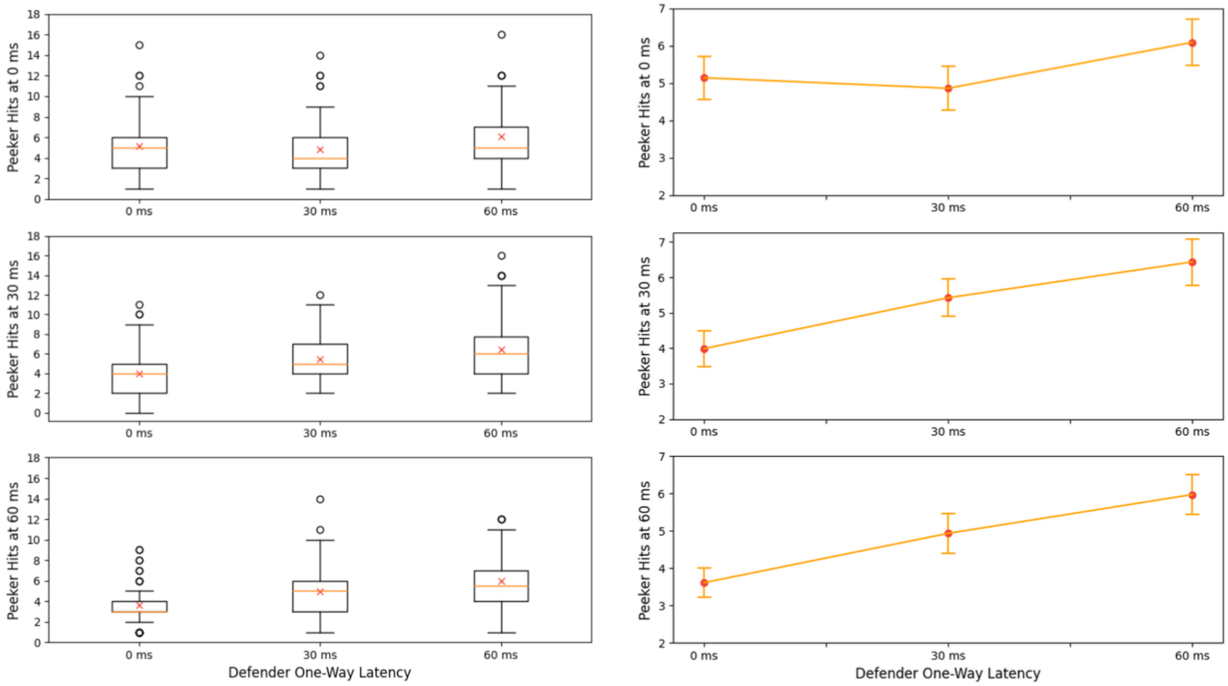
(b) Mean reflex time of all players

Figure 8: boxplots of Player's hours played in FPS games and average reflex time

Most users are experienced FPS players reporting between 200 to 3500 hours. One user has 8000 hours in FPS games but most FPS hours skew towards the low end. Most users have very fast reflex times of around 200 ms, the distribution is symmetrical and there are three outliers.

4.2 The Effect of Latency on Player Performance

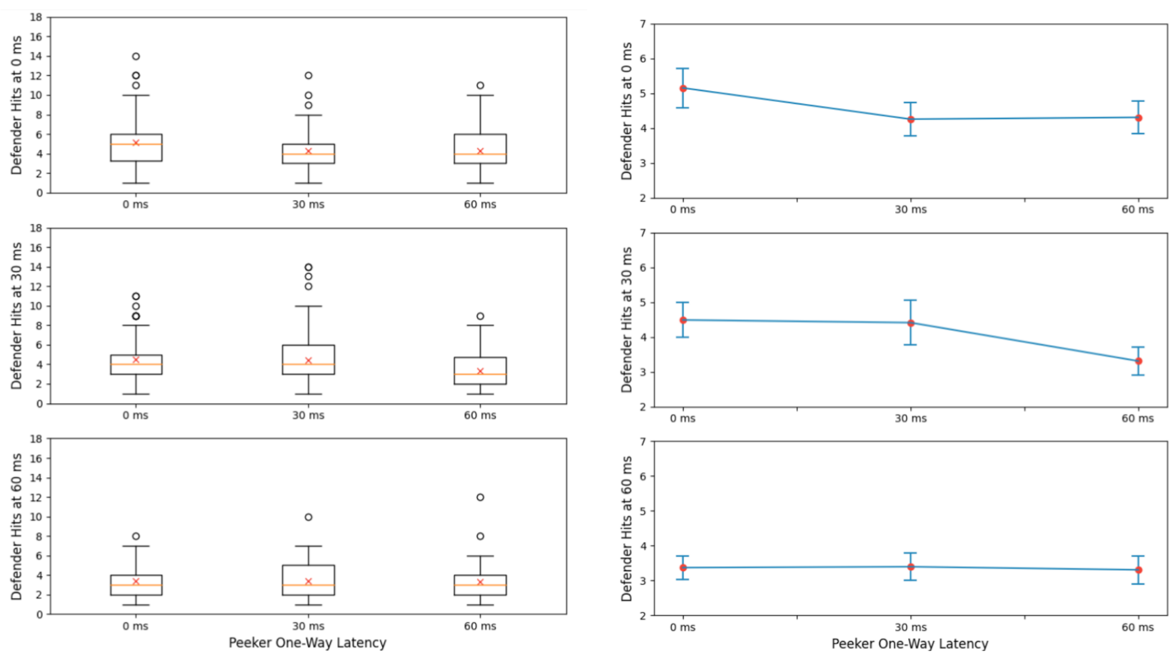
Figure 9a shows the distribution of hits per round for the peeker in box plots. Figure 9b showcases the average hits per round with a 95% confidence interval. Figure 10a shows the distribution of hits per round for the defender. Figure 10b showcases the average hits per round with 95% confidence intervals.



(a) The distribution of hits per round of the peeker

(b) Average hits per round of the peeker.

Figure 9: Peeker hits versus defender one-way latency.



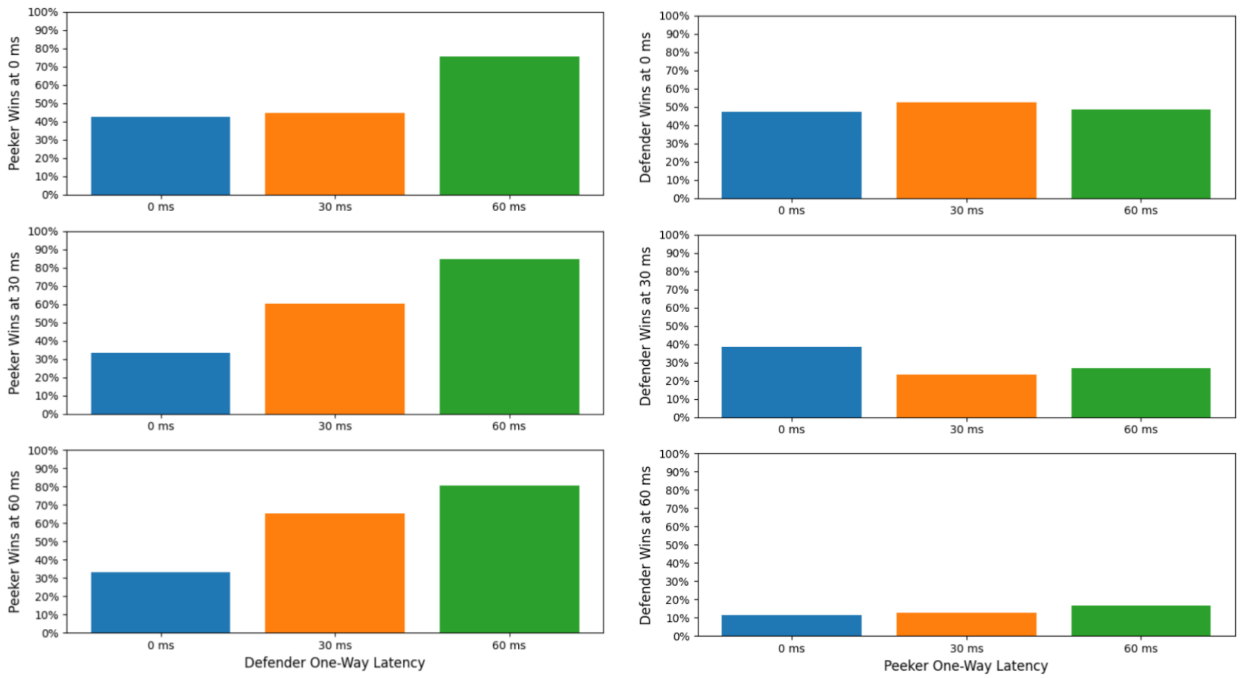
(a) Distribution of hits per round of the defender.

(b) Average hits per round of the defender.

Figure 10: Defender hits versus peeker one-way latency

The peeker's hits increase as the latency of the defender increases. However, the peeker's hits do not vary much with the peeker's own latency. Both roles' latency has effects on the defender's hits except when the defender is at 60ms where the hits are unaffected by the peeker's latency. In every match-up where the defender had 60 ms latency, the peeker has a higher average number of hits per round than when the defender has any other latency, regardless of the peekers own latency

Figure 11a showcases the percentage of rounds won by the peeker when at a certain latency, versus the defender at a certain latency. Figure 11b showcases the percentage of rounds won by the defender when at a certain latency, versus the peeker at a certain latency.



(a) Win percentage of the peeker, with the x axis being the defender's latency.

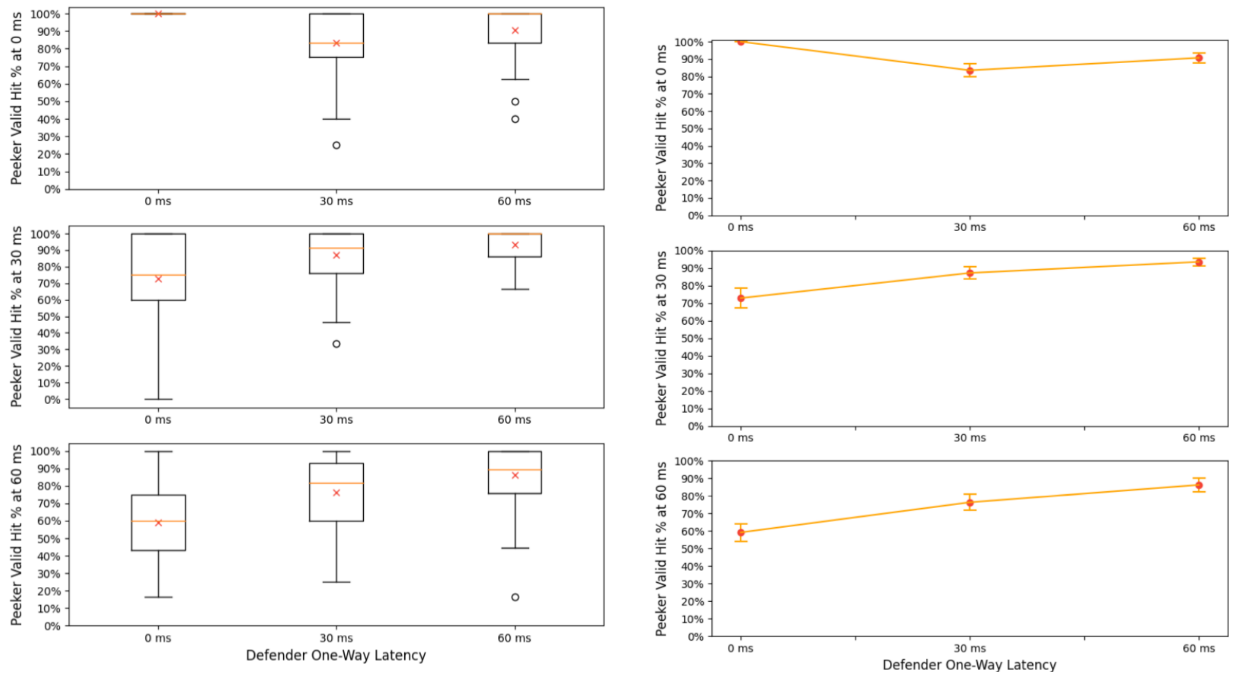
(b) Win percentage of the defender, with the x axis being the peeker's latency.

Figure 11: Win rates

The peeker's win rate is affected by the defender's latency but is not affected by the peeker's latency. However, the defender's win rate is affected by the defender's own latency. The amount of round wins for the defender drops drastically as the latency increases - starting with almost even wins against the peeker at 0 ms latency and dropping to almost never winning rounds at 60 ms latency.

Hits used in the analysis are valid hits. Since FPSci is client authoritative, if one player fires within the network round-trip after the other player, both players' hits are registered. However, in a server authoritative setup, only the hit of the player who shot first is registered. Therefore, we filtered out hits as invalid if they would not be reported in a server authoritative setup.

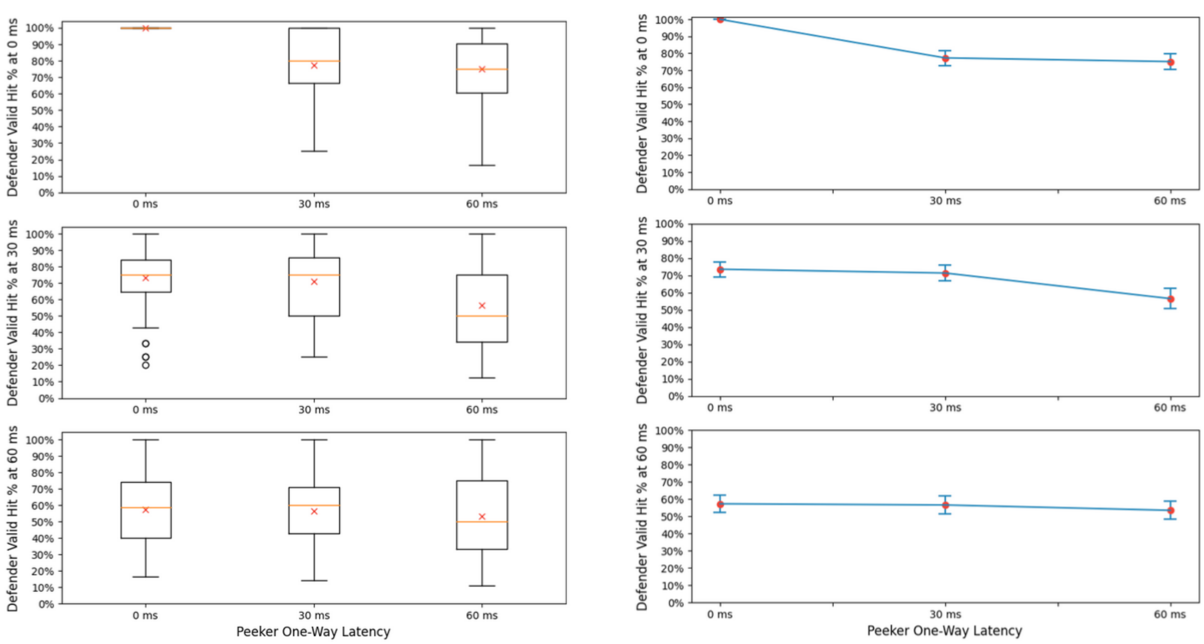
Figure 12a showcases the distribution of the percentage of hits that were valid servicer side out of the total client-side hits. Figure 12b showcases the average percentage of hits that were valid (not within the round-trip time) out of total hits per round (that were calculated client-side) with 95% confidence intervals. Figure 13a showcases the average percentage of hits that were valid out of total hits per round with 95% confidence intervals. Figure 13b showcases the average percentage of hits that were valid out of total hits per round with 95% confidence intervals.



(a) Percentage of peeker hits that were valid out of total hits.

(b) Average percentage of peeker hits that were valid out of total hits.

Figure 12: Peeker valid hits versus defender's latency



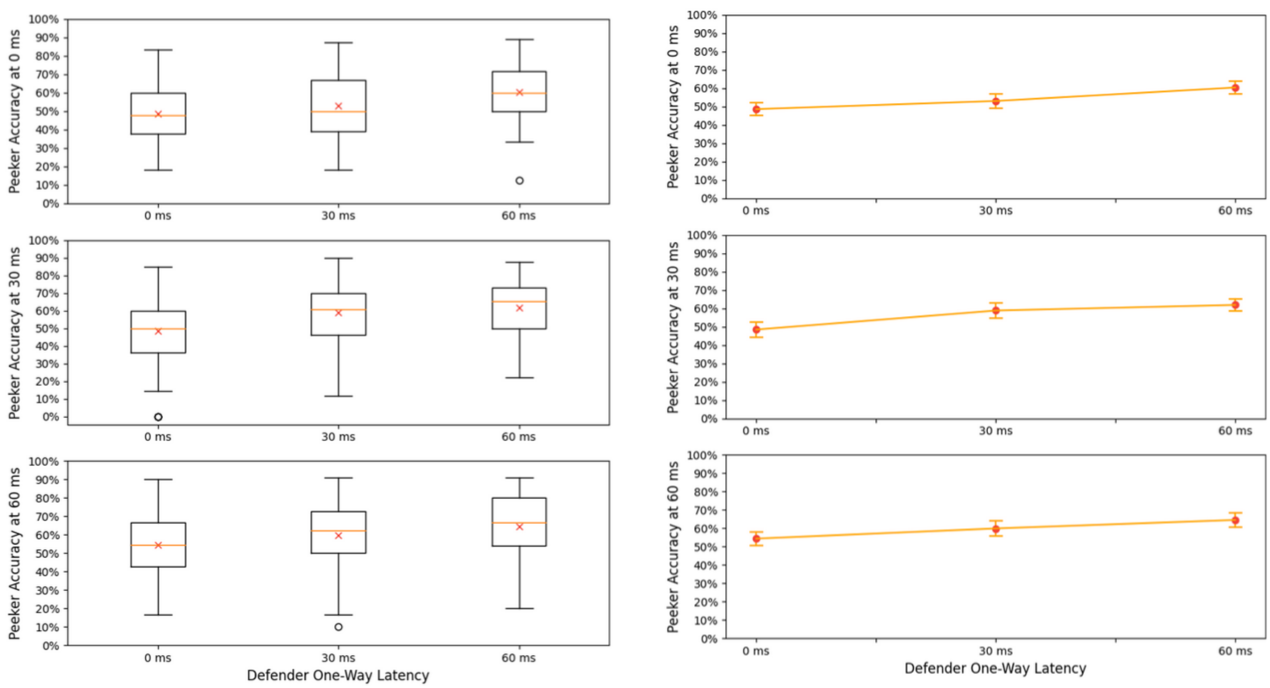
(a) Percentage of defender hits that were valid out of total hits.

(b) Average percentage of defender hits that were valid out of total hits.

Figure 13: Defender valid hits versus peeker's latency.

The results relating to the percentage of valid hits out of total hits tend to favor the peeker. Except for the 0 ms vs 0 ms match wherein every hit was valid due to the lack of latency. The peeker generally had a higher average percentage of valid hits - especially when defenders had a high latency regardless of the peeker's latency. This means that the peeker was likely hitting the defender first more often than not.

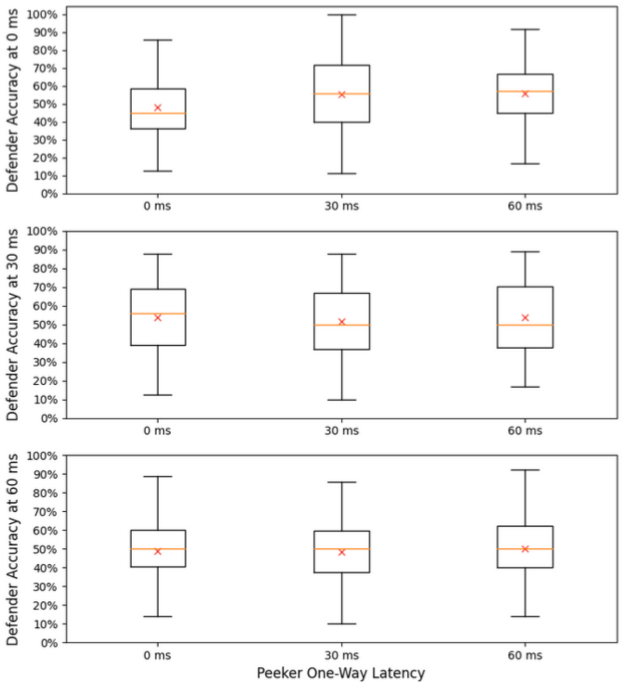
Figure 14a showcases the distribution of the accuracy of the player as the percentage of shots hit out of the total of hits and misses. Figure 14 b showcases the average accuracy per round as a percentage of hits / (hits + misses) with 95% confidence intervals. Figure 15a showcases the distribution of the accuracy of the player as the percentage of shots hit out of the total of hits and misses. Figure 15b showcases the average accuracy per round as a percentage of hits with 95% confidence intervals.



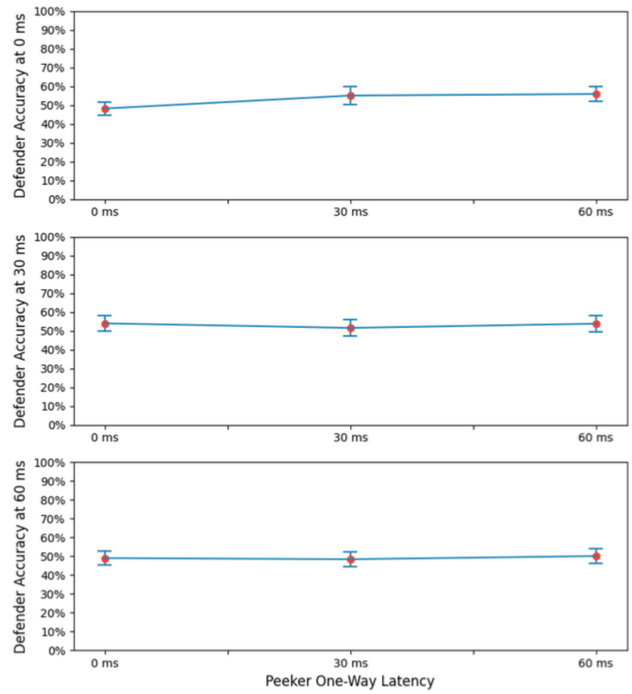
(a) Peeker accuracy as a percentage

(b) Average peeker accuracy as a percentage

Figure 14: Peeker accuracy versus defender's latecy



(a) Defender accuracy as a percentage.



(b) Average defender accuracy as a percentage.

Figure 15: Defender accuracy versus peeker's latency

These graphs show that latency has a small effect on accuracy. While there was some variation between the different latencies, the average accuracy did not change much. Compared to the defender, the peeker's accuracy shows a slight increase as the opponent's latency increases. However, the mean and median variation for both roles at 0 ms and 60 ms are within 10%. In addition, there was a large variation in the distribution of accuracy between players within the same latency.

In summary, the phenomenon known as "Peeker's Advantage" is affected by latency. However, it seems to be far more affected by the defender's latency than the peeker's - in fact, it seems almost completely independent of the peeker's own latency.

4.3 The Effect of Peeker's Distances and Latency on Player Performance

Peeker's distances from the corner were plotted against win rates for both players across all matches in Figure 16. Win rates are round win rates with ties excluded.

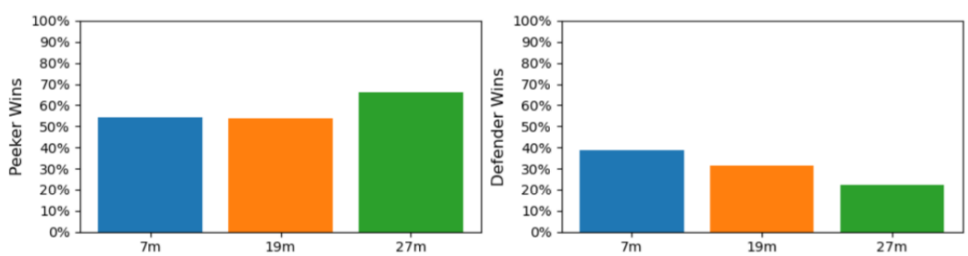
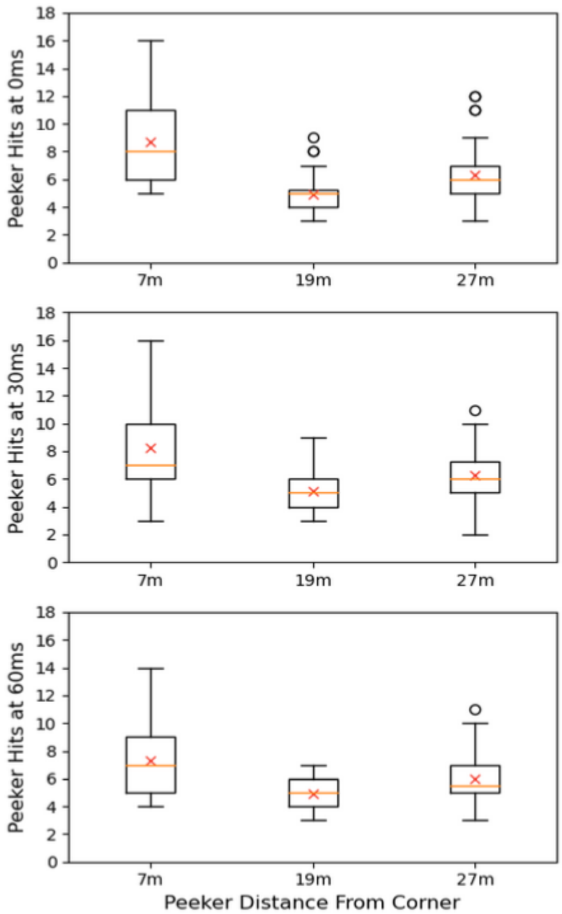


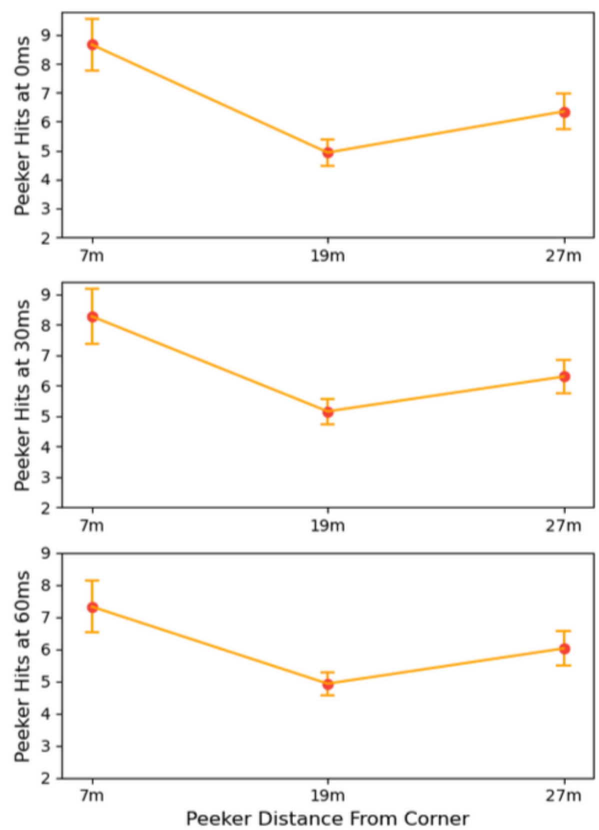
Figure 16: Win rates for both players versus Peeker distance from the corner.

The peeker's average round win rates are both at 53% for 7m and 19m. At 27m, the peeker's win rate increased. The defender's around win rate decreased as latency increased.

Figure 17a shows the distribution of the number of hits by the peeker at the peeker's latency of 0ms, 30ms, and 60ms and at the distance of 7m, 19m, and 27m from the corner. Figure 17b shows the mean number of hits by the peeker under the same conditions. The graphs include mean hits and 95% confidence intervals. Figure 18a shows the distribution of the number of hits by the defender at latencies of 0ms, 30ms, and 60ms and at the peeker's distance of 7m, 19m, and 27m from the corner. Figure 18b shows the mean number of hits by the defender under the same conditions. The graphs include mean hits and 95% confidence intervals.

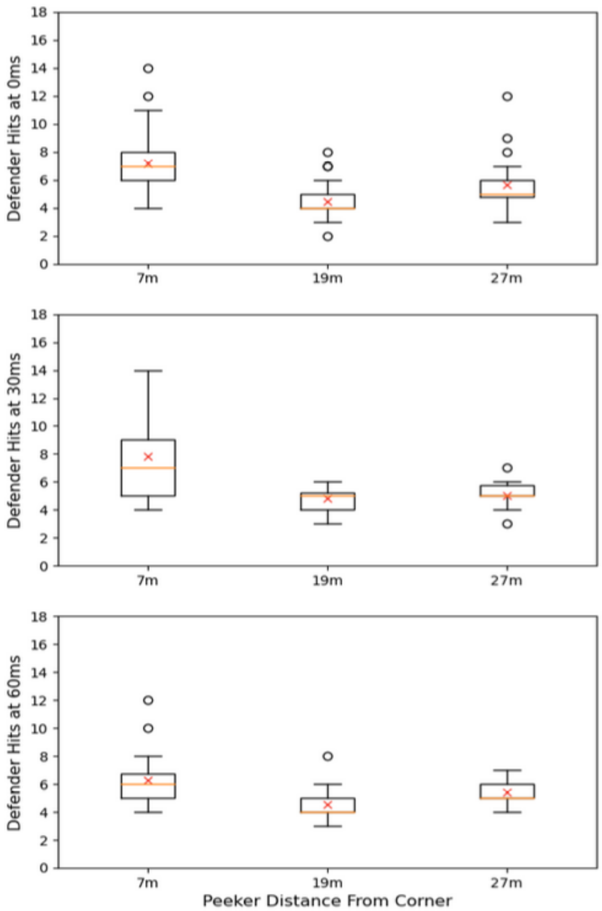


(a) Peeker hits

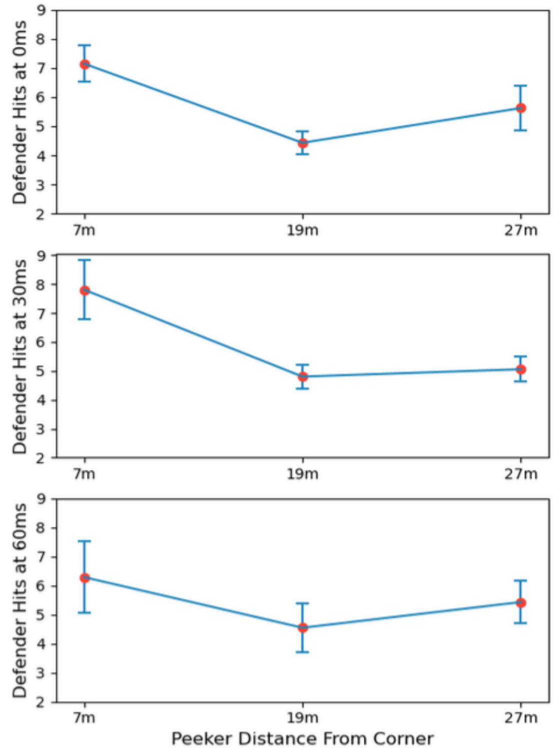


(b) Mean peeker hits

Figure 17: Peeker hits versus peeker distance



(a) Defender hits



(b) Mean defender hits

Figure 18: Defender hits versus peeker distance

By subtracting the mean hits of the defender from that of the peeker at corresponding distances and latencies, we conclude that the peeker's advantage is the greatest at 7m with a mean hits difference of 1.37 across all latency values. This is followed by a mean hits difference of 0.91 at 27m and 0.46 at 19m.

Peeker's Advantage exists at all peeker distances from the corner since the mean and median of the peeker hits at the same peeker's distance under the same latency is always greater than that of the defender. For all three latency values, the defender's and peeker's hit distribution follows almost the same trend: The players hit the most at 7m and the least at 19m, with the number of hits at 27m in the middle. However, when the defender is at 30ms, the distribution and mean of the defender hits at 19m and 27m are very close.

It is worth noting that when both players are at 30ms and when the peeker is at 27m, the peeker has a relatively large advantage of 1.55 hits on average more than the defender.

In summary, the optimal peeking distance is 7m, which is the closest distance used in our experiment. This is contrary to the popular belief one should peek as far away from the corner as possible.

4.4 Time-To-Damage

Time-to-damage refers to the time it takes for a player to hit their opponent after they obtained a line of sight of the opponent. Time-to-damage data was calculated across all rounds for all players. The data was split by player roles into defender and peeker and used to generate the following box and whiskers plot in Figure 19.

In our data, generally the defender has a lower time to damage than the peeker. The median time-to-

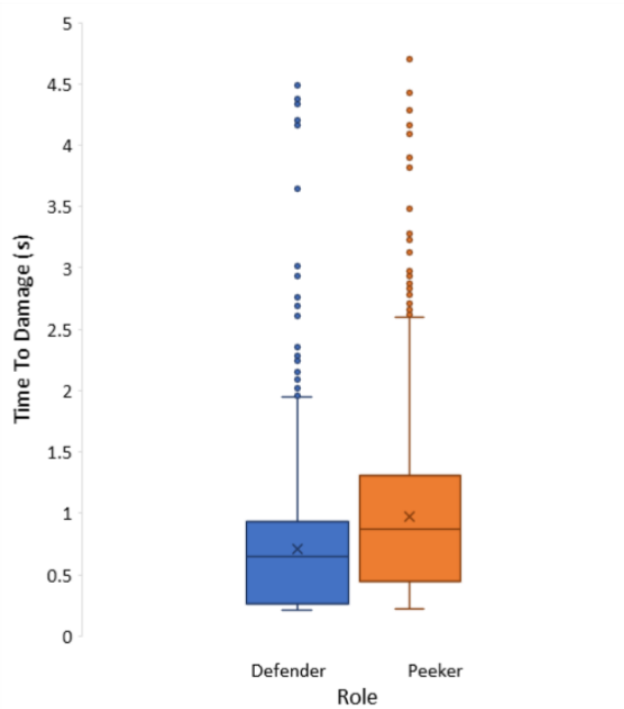


Figure 19: Time to damage boxplot for defender and peeker

damage of the defender is 0.645 seconds and the median time-to-damage of the peeker is 0.871 seconds. The highest outliers for both players are below 5 seconds. These outliers likely indicate that the user misses their first shot.

Overall, the peeker's time-to-damage is higher than the defender's time-to-damage. We theorized that this could be caused by the peeker having to actively search for and aim at the defender, whereas the defender can always aim at the corner and wait for the peeker to appear.

4.5 Reflex Time and Performance

Figure 20 shows the mean reflex time plotted against the corresponding player's mean time to damage. The correlation is weak.

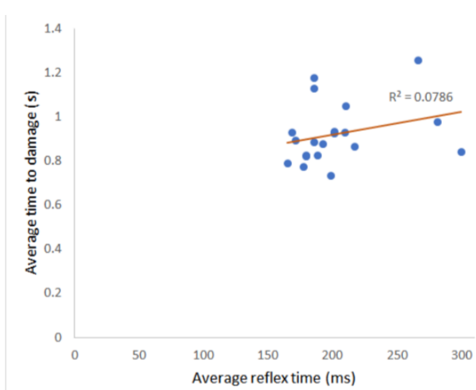


Figure 20: Average time to damage versus average reflex time

Difference in the number of rounds won (winning player rounds won - losing player rounds won) was plotted against the difference in reflex time (losing player reflex time - winning player reflex time). Figure 21 shows a positive correlation between the difference in reflex time and the difference in rounds won.

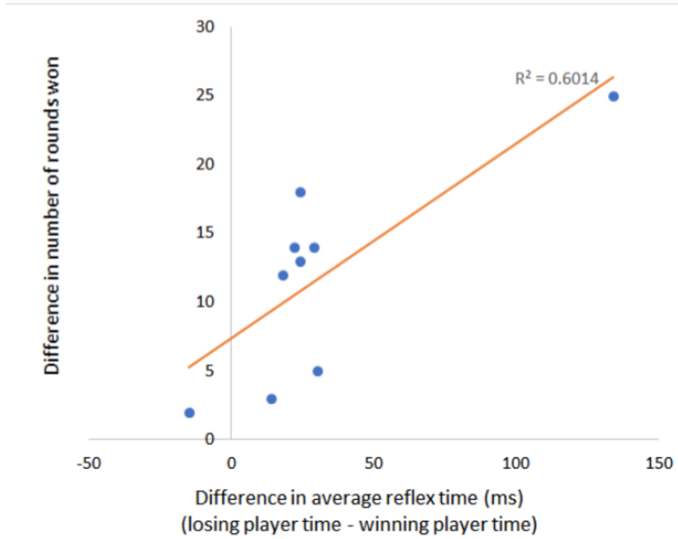


Figure 21: Difference in number of rounds won versus difference in average reflex time

The following plot in Figure 22 was generated without the outlier. It shows a relatively low correlation between the difference in average reflex time and the difference in the number of rounds won.

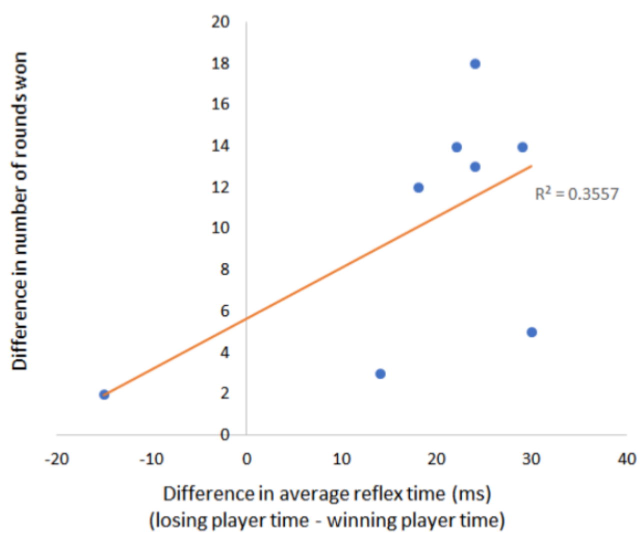


Figure 22: Difference in average reflex time versus Difference in the number of rounds won with the outlier excluded.

In all except one game, the user with the lower reflex time won. In our study, the lower reflex time has a performance advantage. However, time-to-damage does not appear to be influenced by reflex time.

4.6 Quality of Experience

We calculated the mean Quality of Experience (QoE) answers that the players gave for each latency value and created the bar plot shown in Figure 23. Figure 24 and Figure 25 show QoE answers for each role.

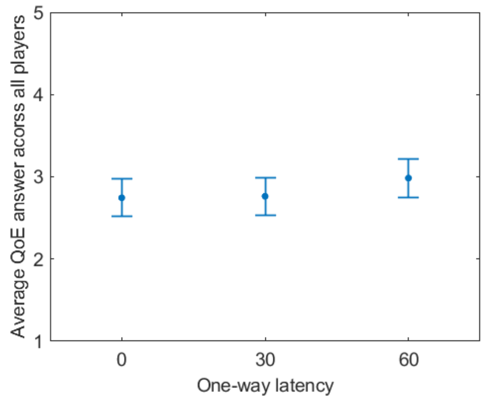


Figure 23: Mean and confidence interval of the five answers across all users.

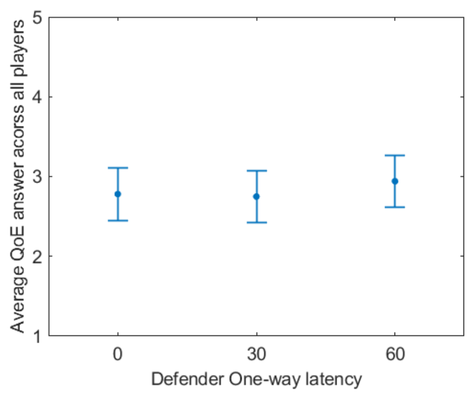


Figure 24: Mean and confidence interval of the five answers that the defender gave for each latency value.

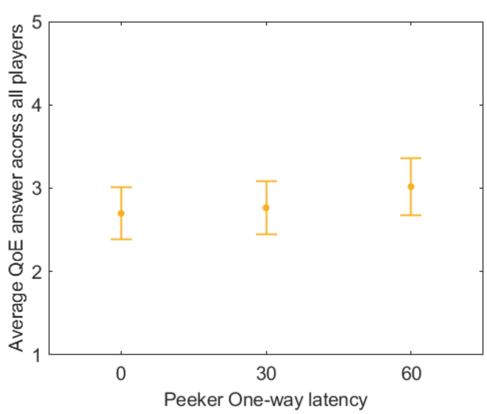


Figure 25: Mean and confidence interval of the five answers that the peeker gave for each latency value.

In general, users gave similar answers for 0 and 30 ms. At 60 ms, on average, the users gave higher QoE answers. The defender's answer follows the same trend. The peeker gave 30 ms a slightly higher score than 0 ms.

4.7 Summary

Our data shows that "Peeker's Advantage" is affected by latency. Doth hits and win rates show a much greater variation when the defender's latency changes than when the peeker's latency changes. Our data support hypothesis 1 outlined in the Introduction chapter: the effectiveness of peeking versus defending is more affected by the defender's latency than by the peeker's.

The number of hits done by both roles changes similarly as latency increases for all peeker distances from the corner. Our data support hypothesis 2 outlined in the Introduction chapter: there is an optimal distance between the peeker and the corner, which does not vary with latency. At 7m, Peeker's Advantage is the greatest and we did not observe changes as latency increases. However, at all latency levels, the relationship between Peeker's Advantage and distance is not linear. The peeker is the least advantageous at 19m, but Peeker's Advantage increases as the distance to the corner increases to 27m.

5 Future Work

Future work includes analyzing the data we obtained from the tournaments, studying the effect of different map geometry on Peeker's Advantage, and studying the effect of game features that were not implemented in our study on Peeker's Advantage,

Data from the tournaments has not been analyzed in our study. Analyzing the tournament's data could be interesting since the player's skills are more closely matched. Also, the effect of stress on players' performances may be stronger in the tournament data. This work can be started by creating graphs of the data using the same methods in this study.

Our study only covered peeking to the right of a 90-degree corner. Altering the map geometry could allow future studies on the effect of map geometry on Peeker's Advantage. A new map with corners for peeking to the left, corners with chamfered edges, or corners that are not perpendicular to the floor could be created.

Game features such as asymmetric latency, latency compensation, and server authoritative architecture have not been implemented in FPSci, therefore were not variables in our study. These features are present in many competitive games. Other root causes of Peeker's Advantage may be found by running a similar study with these features as independent variables.

The addition of inaccuracy caused by player movement can be implemented. In games such as CSGO and Valorant, players are only fully accurate when standing still. This feature can be added to the study and its effect on players' performances can be explored.

6 Conclusion

As FPS games and esports are becoming popular forms of entertainment and a rapidly growing industry, more and more players joining online multiplayer games. Player performance and user experience are impacted by network latency. While many previous studies have focused on the effect of latency on a single player's targeting and navigation ability, the effect of latency in combination with the peeker's distance to the corner on player performance in a peeking and defending scenerio has not been thoroughly studied despite the common belief among players that peeker's advantage is caused by latency. Therefore, the effect of latency and the peeker's distance from the corner on the peeker's advantage is the focus of this study.

A user study was performed using an open-source FPS game: First Person Science developed by Nvidia and modified by the MQP team to support client-server play. Three one-way latency values (0 ms, 30 ms, and 60 ms) and three distances (7 m, 19 m, and 27 m) were used in the study. Users were assigned to the roles of either a peeker or a defender randomly and were placed on a custom map. One latency values was assigned to each user and one out of the three distances from the corner was assigned to the peeker for each round randomly. A second data set was gathered to simulate stress in a real-world esports event, the study was streamed and users were invited back to compete in a tournament for a prize. The reflex time of each user was recorded.

Data were gathered from 24 users and were analyzed for number of hits, win rate, valid hits, accuracy, and time-to-damage. The analysis shows that the peeker's advantage is affected by the defender's latency and is almost not affected by the peeker's latency. Both hits and win rates show a much greater variation when the defender's latency changes than when the peeker's latency changes. Both roles had higher QoE for higher latencies. However, the peeker distinguished lower latencies better since the peeker had higher QoE for 60 ms than 30 ms while the defender had similar QoE for both.

Peeker's advantage is the greatest when the peeker is closest to the corner. Peeker's advantage is the smallest at medium distances and then increases as the distance increases. The optimal distance does not change with latency.

References

[1] "FPS game market growing demand and trends 2023 to 2028," *MarketWatch.* https://www.marketwatch.com/press-release/fps-game-market-growing-demand-and-trends-2023-to-2028-2023-02-27, Feb 2023, last accessed on Mar 5, 2023.

[2] A. Stavropoulos, "Peeker's advantage in Valorant explained," *Dot Esports.* https://dotesports.com/valorant/news/peekers-advantage-in-valorant-explained, Mar 2021, last accessed on Mar 20, 2023.

[3] T. Henderson, "Latency and user behaviour on a multiplayer game server," *Networked Group Communication, Third International COST264 Workshop, London, UK*, p. 1–13, Nov 2001.

[4] C. Jiang, A. Kundu, S. Liu, R. Salay, X. Xu, and M. Claypool, "A survey of player opinions of network latency in online games," *WPI Computer Science Technical Report Series, Worcester, MA, USA*, Jul 2020.

[5] S. Liu, M. Claypool, A. Kuwahara, J. Scovell, and J. Sherman, "L33t or N00b?: how player skill alters the effects of network latency on first person shooter game players," *Proceedings of the Workshop on Game Systems (GameSys), Istanbul, Turkey*, Sep 2021.

[6] S. Liu, A. Kuwahara, J. Scovell, J. Sherman, and M. Claypool, "Comparing the effects of network latency versus local latency on competitive first person shooter game players," *ACM Esports and High Performance HCI Workshop (EHPHCI), Virtual Conference*, May 2021.

[7] S. Liu, X. Xu, and M. Claypool, "A survey and taxonomy of latency compensation techniques for network computer games," *ACM Computing Surveys*, vol. 54, no. 11s, p. 1–34, Jan 2022.

[8] S. W. K. Lee and R. K. C. Chang, "On shot around a corner in first-person shooter games," *15th Annual Workshop on Network and Systems Support for Games (NetGames), Taipei, Taiwan*, Jun 2017.

[9] K. T. Claypool and M. Claypool, "On frame rate and player performance in first person shooter games," *Multimedia Systems*, vol. 13, no. 1, p. 3–17, Apr 2007.

[10] R. Eg, K. Raaen, and M. Claypool, "Playing with delay: With poor timing comes poor performance, and experience follows suit," *Tenth International Conference on Quality of Multimedia Experience (QoMEX), Cagliari, Italy*, May 2018.

[11] M. Claypool, K. Claypool, and F. Damaa, "The effects of frame rate and resolution on users playing first person shooter games," *ACM/SPIE Multimedia Computing and Networking (MMCN), San Jose, California, USA*, Jan 2006.

[12] S. Liu and M. Claypool, "The impact of latency on navigation in a first-person perspective game," *CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA*, May 2022.

[13] X. Xu, S. Liu, and M. Claypool, "The effects of network latency on Counter-strike: Global offensive players," *14th International Conference on Quality of Multimedia Experience (QoMEX), Lippstadt, Germany*, Sep 2022.

[14] A. Hayden and B. Peters, "CS/IMGD MQP - First-Person Science - exploring FPS games and latency," *Worcester Polytechnic Institute.* https://digital.wpi.edu/concern/student_works/6969z402w?locale=en, Dec 2022, last accessed on Mar 22, 2023.

[15] X. Xu, "Reaction time test," *Worcester Polytechnic Institute.* https://web.cs.wpi.edu/~xxu11/, Jul 17, 2020, last accessed on Mar 22, 2023.

[16] X. Xu, "Reaction time test code," *Worcester Polytechnic Institute.* https://drive.google.com/file/d/1u6RbWoHCacfH8RYI9BYJ2O8Z1jjGt0Q0/view, Jul. 17, 2020, last accessed on Mar 22, 2023.