The Effects of Latency on Online Madden NFL Football

James Nichols and Mark Claypool
Computer Science Department at Worcester Polytechnic Institute
Worcester, MA, 01609, USA
{jnick|claypool}@cs.wpi.edu

ABSTRACT

With growth in interactive network games comes increased importance in a better understanding of the effects of latency on game performance. While previous work has measured the effects of latency on first-person shooters and real-time strategy games, there has been no systematic investigation of the effects of latency on sports games. In this work, we study the effects of latency on online Madden NFL football, one of the most popular online sports games, through a series of carefully designed experiments in which we systematically control latency between players. Our experiments illustrate the mechanisms Madden NFL uses to compensate for latency. Our user studies show there is little impact from latency on user performance in Madden NFL over typical low Internet latencies. However, for latencies higher than 500 ms, there is a significant impact on user performance, degrading performance by almost 30%. Our network measurements show periodic data rates during game-play with significant command aggregation at higher latencies.

Categories and Subject Descriptors: C.2.m [Computer-Communication Networks]: Miscellaneous

General Terms: Performance, Design, Human Factors.

Keywords: Network Games, Latency Compensation.

1. INTRODUCTION

In 2002, over 221 million computer and video games were sold, or almost two games for every household in America.[1] Computer games was the only entertainment industry to continue to grow in 2003 [5] and as of the end of 2003, gross revenue from computer game sales surpassed revenues from movie ticket sales, video rentals and concert tickets [4]. The online component of video games has also grown considerably with some games being released with only online multi-player play. Multi-player network computer games can make up around half of the top 25 types of non-traditional traffic for some Internet links [8] and are predicted to make up over 25% of Local Area Network (LAN) traffic by the year 2010.

Knowledge of how network related issues, such as latency and packet loss, affect the usability of games can be of great use to the companies that make these games, network software and equipment manufacturers, Internet Service Providers (ISPs), and the research community at large. Moreover, experimental study of network games can provide the data required for accurate simulations, a typical tool for evaluating network research, as well as insight for network architectures and designs that more effectively accommodate network game traffic turbulence.

While there has been research qualitatively characterizing the effects of latency for car racing [11], custom games [12], first-person shooter (FPS) games [1, 6] and real-time strategy games [13] as well as a general awareness of latency issues [2, 3, 9], quantitative studies of the effects of latency on sports games have been lacking. Moreover, it is unlikely that these other games have the same network requirements as do sports games. For example, in many FPS games, exact positioning and timing is required, because a target must still be at the location where a player aimed in order for a shot to hit. In sports games, the positioning and timing is more forgiving because, for instance, a player cannot kick a soccer ball or throw a football as fast as a bullet.

In this work, we study a sports game in order to begin to fill in the gap in knowledge of the impact of latency on the sports genre. Furthermore, we study game consoles, as opposed to games on a PC, since sports games are far more popular on consoles than they are on PCs [15]. This popularity difference may be caused by the different types of physical user interaction on consoles (which is predominantly with hand controllers) and computers (which is predominantly with mice and keyboards). For our choice of sports game, we examine the popular online sports game, EA Sports' Madden NFL® football.[2] In 2001, EA reported that 200,000 new users registered to play Madden NFL online a few weeks after the game was released, and the 2004 online Madden NFL Website reports thousands of users online on a typical weeknight and 7000 games played per hour.

This paper makes three main contributions to the study of online sports games. First, Section 3 uses three carefully designed experiments to provide evidence for the latency compensation technique used by online Madden NFL football. These experiments can be reproduced by other researchers for other online games to determine how they might compensate for latency. Second, Section 4 presents carefully

[1] Top Ten Industry Facts, IDSA, http://www.idsa.com/-pressroom.html

[2] http://www.easports.com/games/madden2004/home.jsp

NOSSDAV'04, June 16–18, 2004, Kinsale, County Cork, Ireland.

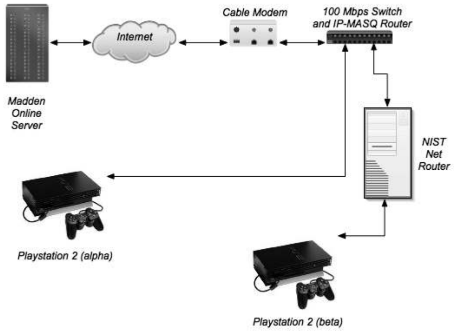

Figure 1: Experimental Testbed.

designed users studies that identify how latency affects running and passing, two fundamental interaction components in football. And third, Section 5 analyzes network level data for online Madden NFL football, showing how latency affects packet sizes and data rates.

2. NETWORK GAME TESTBED

We constructed a testbed that allows systematic control of latency for a two-player console game. The testbed, depicted in Figure 1, contains two Sony PlayStation® 2 consoles (labeled *alpha* and *beta*), each running the 2004 edition of Madden NFL football. Both consoles are located on the same Ethernet segment, with console Beta behind a proxy-ARP router. The proxy-ARP router runs the NIST Net network emulator, a Linux kernel module that allows allows us to induce latency on packets to and from console Beta. The online Madden NFL server is not used during the actual game play itself, but rather simply serves to facilitate people finding each other before games start. Periodically during game play, each console does send a few packets of data to the online Madden NFL server, but this is merely to update the online status for other users who may be interested in finding particular people.

During an online game, traffic is sent from each console through the switch to the IP masquerading router's external IP address. When the traffic reaches the router, it modifies the addresses as appropriate and re-routes the traffic back through itself to the appropriate console. Ping packets sent from the router to the console show the router and switch add less than 5 ms or round-trip latency.

Finally, we connect each console into separate inputs on a single television, allowing us to do picture-in-picture to simultaneously see what each console is displaying.

3. LATENCY COMPENSATION

Online game systems can attempt to compensate for the impact of Internet latencies with various latency compensation techniques [14]. Understanding the latency compensation technique of an online game is a necessary first step in understanding the impact of latency on that game. We determine the latency compensation techniques used by online Madden NFL football through three simple experiments.

In the first experiment, referring to the names for the PlayStation 2 consoles denoted in Figure 1, Beta "challenges" Alpha through the online Madden NFL interface. We then induce a large delay of 1500 ms from Beta to Alpha. Alpha starts on offense and puts an offensive player in motion to have the player move before the play starts. The result is that Beta sees the in-motion player movement first, and subsequently, the player is one or two steps ahead on Beta's display than it is on Alpha's. In other words, Alpha's display lags that of Beta's. Figure 2 shows the results of this experiment. Beta's display is the larger picture, while Alpha's display is inset in the picture-in-picture. Figure 3 shows Alpha's display enlarged, which is somewhat blurry because we are zooming in on the typically coarse television resolution of a picture-in-picture. We have drawn a box around the man in motion on each display to indicate the player of interest. Notice how the boxed player in Figure 3 is further to the left than the boxed player in Figure 2. Similarly, if Beta moves a defensive player, Beta sees it immediately, while Alpha's display is lagged. We see similar phenomena for other aspects of game play, including when Beta is on offense, or for the fair-catch indicator during kicks.

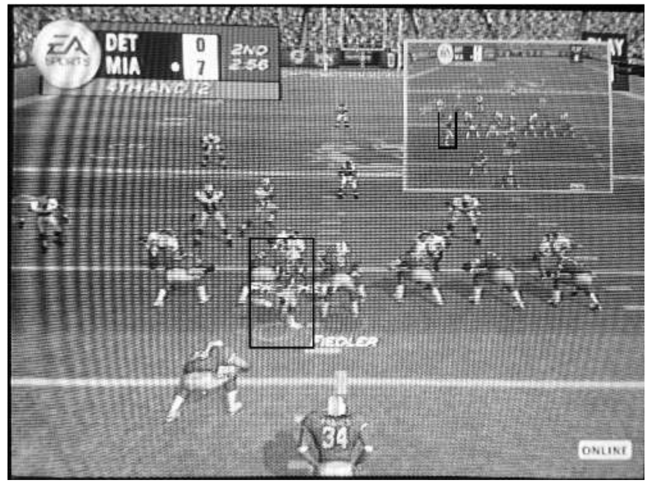

Figure 2: Beta's display with Alpha's inset.

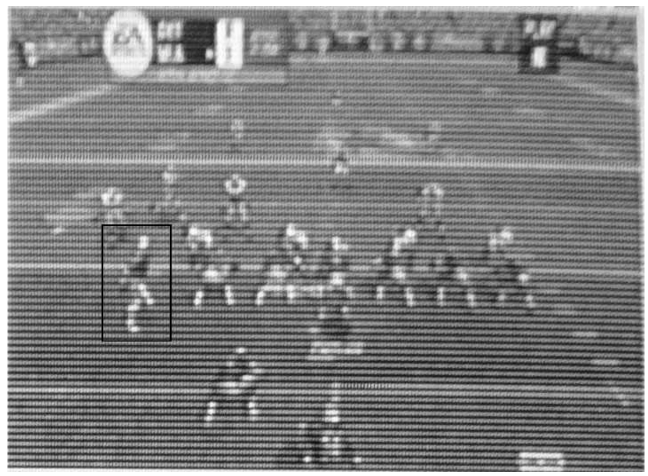

Figure 3: Alpha's display enlarged.

That Alpha waits to render the player movement suggests that online Madden NFL football may be using a "dumb-

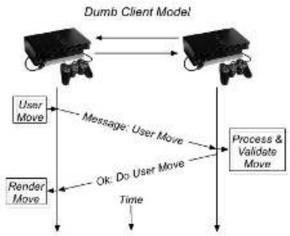

Figure 4: Dumb-Client model.

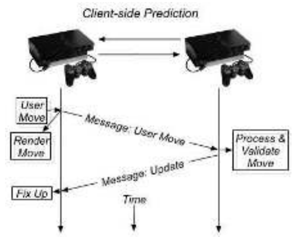

Figure 5: Client-side prediction model.

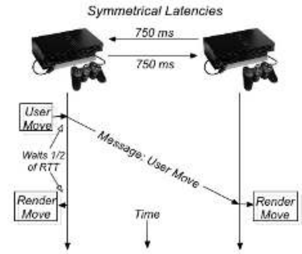

Figure 6: Inferred latency compensation technique used by online Madden NFL football.

client" client-server model [2] used in early network games and depicted in Figure 4. Note, the client-server terminology may be confusing, since examination of the network traffic of Madden NFL football shows a peer-to-peer architecture. For ease of discussion, we consider the client to be where the user input is taking place (Alpha, in the first experiment). In the dumb-client model, the client sends a message to the server when user input is received. The server processes (and validates) the input and sends the results back to the waiting client to render on the local display. Thus, movement is lagged by the round-trip latency between client and server. However, our second experiment reveals that the dumb-client model is *not* used by online Madden NFL football.

In our second experiment, we run the exact same experiment with Beta challenging Alpha except that we reverse the induced latency to be 1500 ms from Alpha to Beta. Here, when Alpha is on offense and puts a man in motion, Alpha sees the movement early, while Beta's display is lagged. When Beta moves a defender, Beta's display is again lagged. Thus, Alpha and Beta's displays in Figures 2 and 3 are reversed when the latency is reversed.

This second experiment suggests that online Madden NFL football is using "client-side prediction". In client-side prediction the local game client instantly responds to user input and renders player movements, then sends a message to the other game participants notifying them of the user input [2]. A diagram of client-side prediction is shown in Figure 5. When the remote software receives the message it renders the player movement on the local display and the user watching this display can then respond appropriately. Thus, remote player actions are lagged slightly on the local host. However, with client-side prediction, in the first experiment, the player on Alpha's display would have started movement first, then a short time (the fundamental latency on the testbed) after the player on Beta's display would have started movement. Instead, the movement of Alpha's player was lagged, while Beta's player moved first. Thus, while client-side prediction explains the results of this second experiment, it does *not* explain the results in the first experiment.

In our third experiment, Beta challenges Alpha and we set 750 ms of latency in both directions between Alpha and Beta. For all cases in this third experiment, player movements are visually synchronized on both Alpha's and Beta's displays.

The results of this third experiment, combined with the results of the first two experiments, suggest an alternate latency compensation technique used in online Madden NFL football, depicted in Figure 6. Upon user input, the local

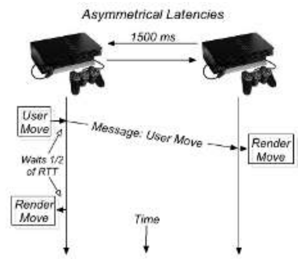

Figure 7: First experiment.

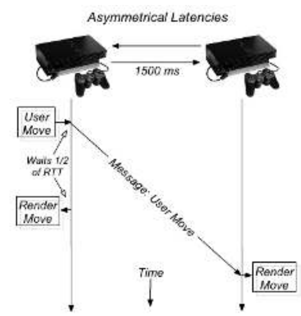

Figure 8: Second experiment.

client console sends a message to the remote console notifying it of the input. After sending this notification the local console waits for 1/2 of the estimated round-trip time before rendering the player movement, assuming that at approximately 1/2 the round-trip time the user input notification message will reach the remote console. Immediately upon receiving the user input message, the remote console renders the player movement. With symmetrical latencies on the link, such as in experiment three, both the local and remote displays are approximately synchronized, even at very high latencies.

This latency compensation technique also explains the results seen in experiments one and two, as shown in Figures 7 and 8. In the first experiment, Alpha processes the user input and waits for 1/2 of the estimated round-trip time (approximately 750 ms) before rendering the player movement. However the user notification reaches Beta in just a few milliseconds which results in Alpha's display being lagged behind Beta's. The converse is evidenced in the second experiment, where Alpha waits 1/2 of the estimated round-trip time before rendering the player movement, but the notification message reaches Beta after 1500 ms, causing Beta's display to be lagged behind Alpha's. This latency compensation technique may be effective for symmetric latencies, but, based on the inconsistent states on each display for experiments one and two, fails when link latencies are asymmetrical.

4. IMPACT OF LATENCY ON USER PERFORMANCE

Through pilot studies and hours of play-testing, we choose to focus on the effect of latency on the two fundamental offense components: running and passing. Next, we deter-

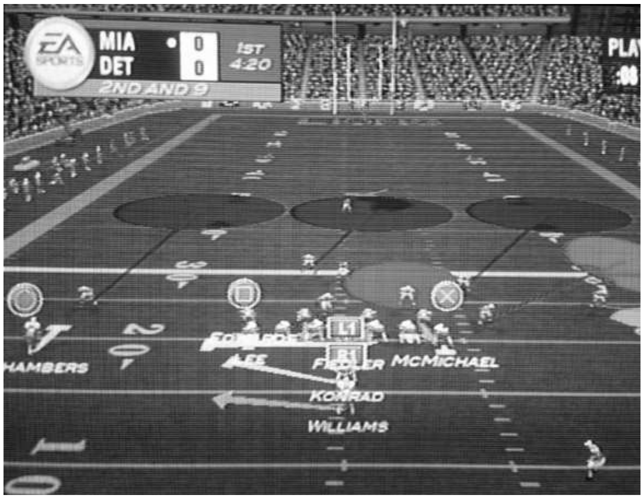

Figure 9: Offense and Defense plays illustrated.

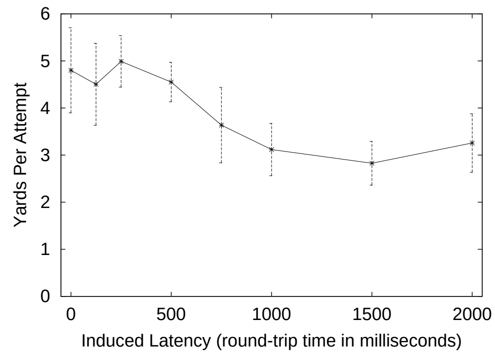

Figure 10: User Run Performance versus Latency.

mine ways to quantitatively measure user performance in regards to these interactions. Since statistics are an integral part of sports, Madden NFL football records a variety of application performance statistics. We select *yards per attempt* as a fundamental measure of running performance and *completion percentage* as a fundamental measure of passing performance.

We would like to focus on running and passing independently in order to isolate the effects of latency on each. Unfortunately, unlike in other studies that used custom maps to isolate user interactions [13], Madden NFL football has no "maps" and the game incorporates many non-deterministic components from play to play: receivers run slightly different routes, defensive lineman rush the quarterback differently, offensive linemen block differently, players get fatigued for the next play, etc. For example, during a run up the middle of the field, the offensive linemen may clear a hole in the defense for the running back on one play while getting flattened by the defense on the next play, even with the exact same play selection, making it difficult to attribute degradation in run performance solely to latency. These game play components, while realistic, also make it difficult to reproduce interaction scenarios repeatedly.

To best isolate the performance of the user during running we force the defense to pick a play to one side of the field, both in the formation (where the players are at the start of the play) and in coverage (where players move when the ball is snapped). Then, we have the offense run the ball to the opposite side of the field. The plays are illustrated in Figure 9, with most defenders on the right side of the field and the running play going to the left, indicated by the two arrows for the blocking and running back. The user[3] simply tries to gain as much yardage as possible.

Our experiments to evaluate the impact of delay on user performance for running consisted of playing 3 full games at 8 different latencies for a total of 24 data points. The user was subjected to induced latencies ranging from 0 to 2000 ms total round-trip. Since this range is even broader than typically found on the Internet [7] we concentrate our experiments in the range 0 to 500 ms. We shuffle induced latencies from experiment to experiment in attempt to avoid any recency affects.

Figure 10 depicts the experimental results, plotting the average of the average yards per attempt for each game versus the induced latency, with the standard deviation of each average shown with error bars. Over the full range of latencies studied, there is a decrease in performance of about 30% compared to performance with no induced latency. The correlation coefficient for average yards per attempt versus latency is a pretty strong -0.86, but the relationship may not be linear based on the visual curvature. Over the range of latencies typically found on the Internet (below 500 ms) there is not much effect on user performance.

While carrying out this experiments, we were also able to make some observations about the qualitative effect of the latency on user performance. First, round-trip latency and changes in round-trip latency at or below 500 ms are not noticeable to the user. Only after about 750 ms or more of latency is round-trip latency noticeable in that the game feels "laggy." This could explain the relatively flat part of the left side of the curve in Figure 10. Anecdotally, if we add 500+ ms of induced latency during the middle of a play the lagginess is almost immediately perceptible suggesting that the game quickly adapts to changes in latency. Second, while playing a game at higher latencies (750+ ms) the movements of the player are lagged momentarily behind user input, making it hard to accurately time running moves such as spins, jukes, and stiff-arms to avoid the defenders. Third, at high round-trip latencies, occasionally a user makes "mistakes" that are unintentional, such as running out of bounds or running directly into a defender because the actions of the player are not as fast as the user reactions. We used the instant replay feature of Madden NFL football to take a few pictures to illustrate this third phenomenon.

In Figure 11, the running back is running towards the left side of the field to avoid the defender. In Figure 12, the user sees that there is an open lane along the sideline and pushes the controller up to run between the defender and the sideline. However, because of the latency, the processing of this input is delayed so that the command is actually processed after the runner goes out of bounds, as in Figure 13. Because of the latency, the user failed to gain as many yards on this attempt as s/he would have if there was no latency.

[3]The users for all our tests were experienced Madden NFL players.

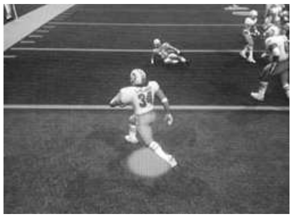

Figure 11: User is pressing left. Player moves left.

Figure 12: User is pressing up. Player continues left because of latency.

Figure 13: Running back goes out of bounds! User curses.

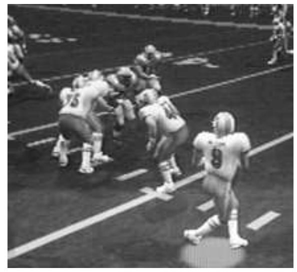

Figure 14: User is pressing throw. Throw is not processed yet because of latency.

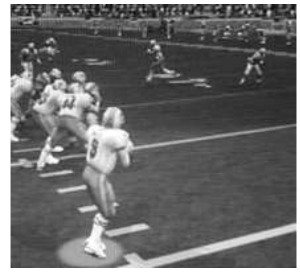

Figure 15: Throw starts processing here because of the latency.

Figure 16: Defender intercepts throw! User curses.

We next investigate the effect of latency on user performance during passing. Our pilot studies with a variety of passing plays suggest latency may have an even larger impact than on running since timing is critical for effective passing. A receiver might only be away from a defender ("open") for a short window of time, perhaps right after executing a particular pass route, making precise timing critical. A good example of this is the "quick slant" passing route, where the receiver quickly runs at a slight angle to the line of scrimmage. The goal of the quick slant route is to catch the defense patrolling certain areas of the field (a "zone" defense) so the quarterback can pass the ball to the receiver on the boundary between two defender areas. Proper timing is essential if the receiver is to catch the ball on this boundary.

Figure 14 depicts the start of play, where, as the receiver begins his route, the user presses the pass button in order to time the pass to reach the receiver at the boundary between defenders. Figure 15 shows where the receiver should be catching the ball at the boundary since he is open. However, due to the latency, the processing of the quarterback throwing the ball actually begins here. By the time the ball reaches the receiver, the receiver has fully crossed the boundary and the defender catches the ball instead (an "interception"), as shown in Figure 16.

We have additional experiments that attempt to precisely quantify the timing aspects critical to passing, but cannot present the results here due to space constraints. We refer the interested reader to [10].

5. NETWORK-LEVEL MEASUREMENTS

Among other things, a better understanding of network game traffic can help design networks and architectures that more effectively accommodate network game traffic patterns. Furthermore, careful empirical measurements of network game traffic can provide data required for accurate simulations, a typical tool for evaluating network research. To better understand network traffic for online Madden NFL football, we run controlled experiments with and without symmetrically inducing a round-trip latency of 1000 ms, capturing all packets on the Ethernet segment after the NIST Net router, on the side closest to Beta in Figure 1. We choose such a large latency to make any effects on the network traffic more evident. For both latency cases, the offense first executes two running plays, then two passing plays, and finally kicks the ball to the defense.

Figure 17 and Figure 18 show the bitrate versus time for the five plays with no induced latency and with 1000 ms induced latency, respectively. The traffic to and from Alpha and Beta is roughly symmetric, as expected given the peer-to-peer architecture in use. We can clearly see five low bitrate periods that correspond to play selection between play action. The overall average bitrate is low, less than 20 Kbps, which further emphasizes that low latency is more important than high capacity for online games. The average bitrate is similar for both the 0 ms and the 1000 ms cases, but the cyclic nature of play action and play selection is more pronounced in the 1000 ms case, perhaps caused by command aggregation.

Figure 19 shows a cumulative density function (CDF) of the packet burst length, which we define as the number of packets that arrive within 15 ms of each other. The steep line at 1 indicates that online Madden NFL does not send traffic in bursts. This is emphasized in Figure 20, which shows that the line for packet sequence number versus time (for a small portion of the traffic from Beta to Alpha) is approximately linear. Although Figures 19 and 20 are for 0 ms induced latency, the results are nearly the same for 1000 ms induced latency.

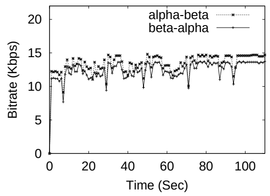

Figure 17: Bitrate versus Time with no induced latency.

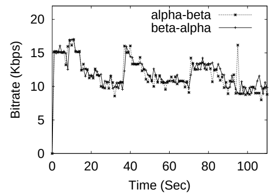

Figure 18: Bitrate versus Time with 1000 ms induced latency.

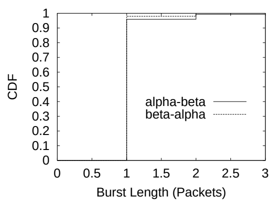

Figure 19: CDF of Burst Length with no induced latency.

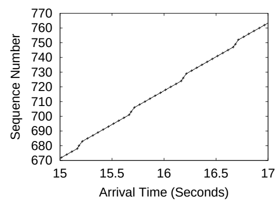

Figure 20: Packet Sequence Number versus Arrival Time with no induced latency.

Figure 21 shows CDFs of inter-arrival times of packets sent from Beta to Alpha for both 0 and 1000 ms induced latency. The CDF distribution shifts for the higher latency and the inter-arrival times vary more widely for the higher latency. Figure 22 shows a corresponding CDF of packet sizes aggregated for packets sent in both directions. The CDF packet size distribution shifts substantially for the higher latency. This is not an artifact of the NIST Net router since it imposes delay symmetrically. With no induced latency, all of the packets are less then 90 bytes and have a median of about 77 bytes. However, for the 1000 ms round-trip time, 90% of the packets are larger than 90 bytes and have a median of about 112 bytes. This suggests online Madden NFL does some command aggregation in the presence of higher latency, which results in larger packet sizes and longer gaps between packet arrivals.

6. CONCLUSIONS

Our experiments suggest that online Madden NFL football uses a prediction of the round-trip time to delay user input in an attempt to compensate for any latency effects across both players. This technique, while effective for symmetric latencies, fails in the presence of asymmetric latencies. Our experiments with users indicate there is little impact from latency on user performance during running for typical Internet latencies, with latencies as high as 500 ms being unnoticeable. However, with latencies higher than 500 ms, running performance can degrade by almost 30%. Overall, we surmise latency artifacts from asymmetric connections are typically dwarfed by the importance of proper play selection; choosing the offensive formation and play execution is more important than occasionally failing to gain all of the available yards on a running play. Based on these preliminary measurements, we suggest online football be placed in a latency QoS category less strict than that for first person shooter games but perhaps in a class more strict than that proposed [13] for real-time strategy games.

Our ongoing work is to determine more effective ways to evaluate latency on passing performance. Evaluation of the impact of other network parameters, such as packet loss may, also help better understand the Quality of Service requirements for online football. Finally, we suggest further investigation other types of sports games, such as soccer to determine their susceptibility to latency.

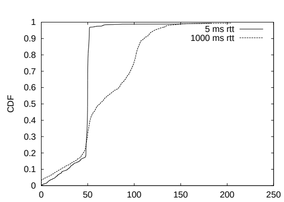

Figure 21: CDF of Inter-arrival Times.

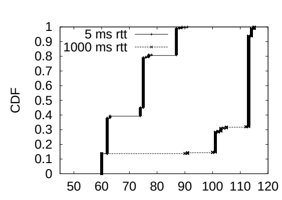

Figure 22: CDF of Packet Sizes.

7. REFERENCES

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

[2] Y. W. Bernier. Latency Compensating Methods in Client/Server In-game Protocol Design and Optimization. In *Proceedings of the Game Developers Conf.*, Feb. 2001.

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

[4] K. Dean. Academic Games Can be Fun. *Wired News*, Nov. 2003.

[5] ESA. Computer and Video Game Software Sales Break $7 Billion in 2003, Jan. 2004. Press Release.

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

[7] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley. Inferring TCP Connection Characteristics Through Passive Measurements. In *Proceedings of IEEE Infocom*, Apr. 2004.

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

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

[10] J. Nichols and M. Claypool. The Effects of Latency on Online Madden NFL Football. Technical Report TR-04-06, CS Department, WPI, Mar. 2004.

[11] L. Pantel and L. C. Wolf. On the Impact of Delay on Real-Time Multiplayer Games. In *Proceedings of ACM NOSSDAV*, May 2002.

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

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

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

[15] The Entertainment Software Association. Essential Facts About the Computer and Video Game Industry, 2003. Press Release. http://www.theesa.com/EF2003.pdf.