The Effects of Latency on Online Madden NFL® Football

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

Abstract— **With the 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 the 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 typically 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.**

I. Introduction

In 2000, the U.S. economy grew only 7.4% while the computer and video game industry grew 14.9%, out-pacing growth in other high-tech industries and even Hollywood over the previous five years [Int01]. In 2002, over 221 million computer and video games were sold, or almost two games for every household in America.[1] The online component of video games has also grown considerably. For example, of the 1.3 million PlayStation 2 owners that bought SOCOM, 400,000 play regularly only online [egm03]. Multi-player network computer games can make up around half of the top 25 types of non-traditional traffic for some Internet links [Mkc00] 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. In particular, if established latency requirements and any associated trade-offs were known, ISPs could establish tariffs based on customers' indicated maximum delays, requested Quality of Service (QoS) and the ISP's ability to meet these demands. Moreover, experimental study of network games can provide the data required for accurate simulations, a typical tool for evaluating network research, as well as insight for network architectures and designs that more effectively accommodate network game traffic turbulence.

While there has been research qualitatively characterizing the effects of latency for car racing [PW02], custom games [SERZ02], popular First-person shooter (FPS) games [Arm03], [Arm01], [Hen01] and real-time strategy games [SGB$^+$03] as well as a general awareness of latency issues [Ber01], [BT01], [Lin99], [Ng97], 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 the player aimed in order for the 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 [The03]. 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 current and perhaps all-time most popular online sports game, EA Sports' Madden NFL® football.[2] The Madden NFL series was started in 1988, and in 2003 was the third top selling video game [The03]. The 2004 edition of Madden sold 2 million copies three weeks after release [Jus03]. Madden NFL was also the first football game to be played online in 2001, and EA reports that 200,000 new users registered to play online weeks after the game was released. The 2004 online Madden NFL Website[3] 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, in Section III, through three carefully designed experiments, this paper provides 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, in Section IV, this paper presents carefully designed users studies and analysis that quantify how latency affects running and passing, two fundamental interaction components in football games. And third, in Section V, this paper analyzes network level data for online Madden NFL football, showing how latency affects packet sizes and data rates.

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

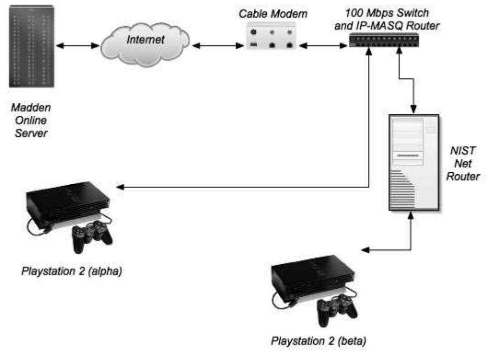

Fig. 1. Experimental Testbed.

II. 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[4] 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 users finding each other before games start.[5]

During an online game, traffic is sent from each console through the switch to the 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 of base 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.

III. Latency Compensation Technique

Online game systems can attempt to compensate for Internet latencies with various latency compensation techniques [SKH02]. Understanding the latency compensation technique of an online game is the 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 PlayStation 2 names 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.[6] 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.[7] We have drawn a box around the man in motion on each display in Figure 2 to indicate the player of interest. 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[8] indicator during punts.

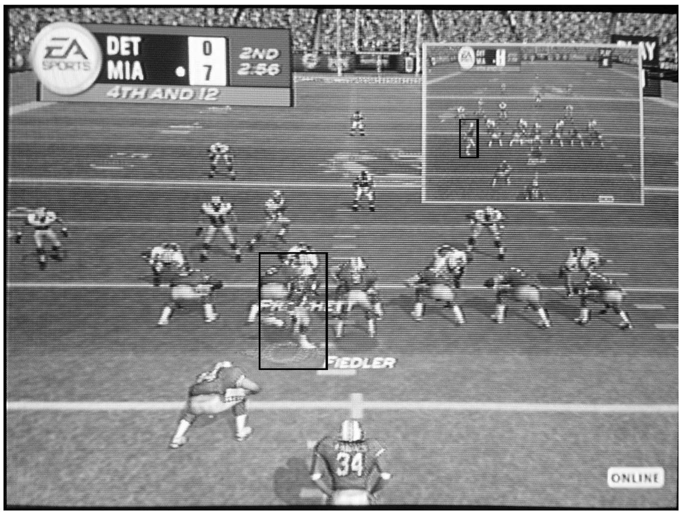

Fig. 2. Beta's display with Alpha's inset.

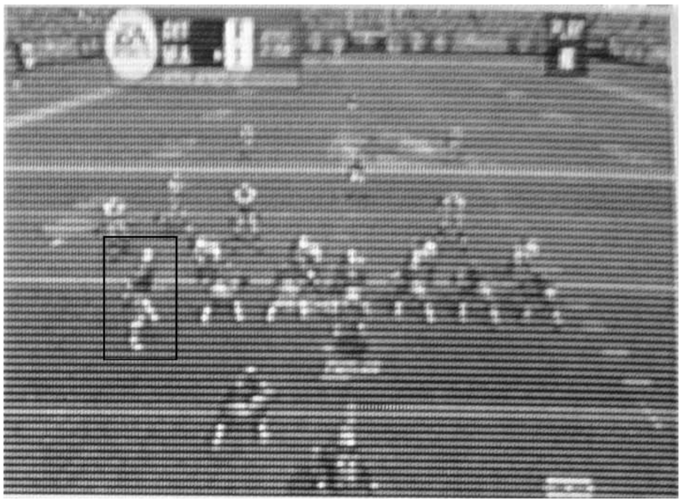

Fig. 3. Alpha's display enlarged.

That Alpha waits to render the player movement suggests that online Madden NFL football may be using a "dumb-client" client-server model [Ber01] used in early network games and depicted in Figure 4.[9]. In the dumb-client model, the client sends

[2] http://www.easports.com/games/madden2004/home.jsp
[3] http://www.easportsonline.com/games/madden2004/home.jsp
[4] http://snad.ncsl.nist.gov/itg/nistnet/
[5] 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.
[6] *In motion* refers to the movement of an offensive player before the play starts.
[7] It is blurry because we are zooming in on the typically coarse television resolution of a picture-in-picture.
[8] A *fair-catch* is when a player indicates they will catch a kicked ball without running.
[9] Client-server terminology may be confusing, since examination of the network traffic of Madden NFL football shows a peer-to-peer architecture. For the

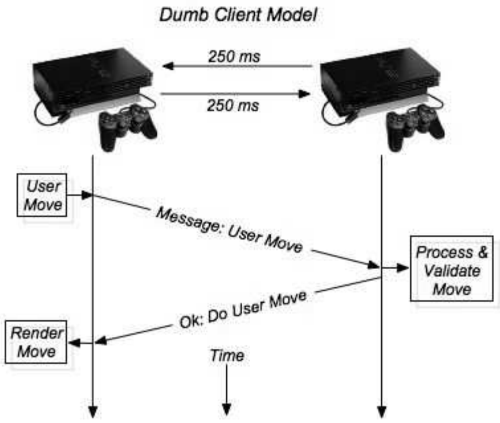

Fig. 4. Dumb-Client model.

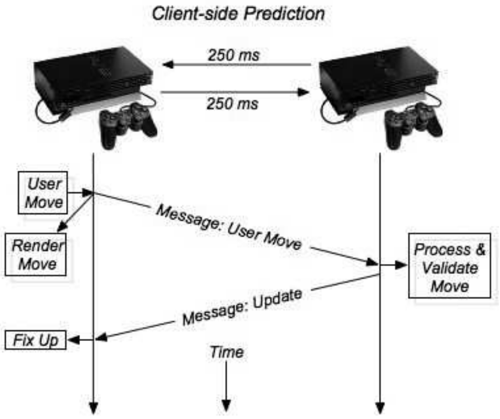

Fig. 5. Client-side prediction.

a message to the server when user input is received. The server process (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 Madded 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. The results are that 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 [Ber01]. 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* appear to be taken in online Madden NFL football based on the first experiment.

sake of discussion, we consider the client to be where the user input is taking place (Alpha, in the first experiment)

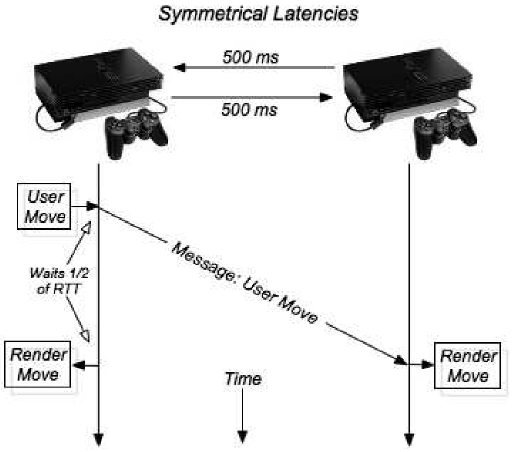

Fig. 6. Inferred latency compensation technique used by online Madden NFL football.

In our third experiment, Beta challenges Alpha and we set 750 ms of latency delay in both directions between Alpha and Beta. For all cases in this third experiment, player movements are roughly 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 client console sends a message to the remote console notifying it of the input. After sending this notification the client console waits for 1/2 of the estimated round-trip time before rendering the player movement, assuming that at approximately 1/s 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, the result is that 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 milliseconds, causing Beta's display to be lagged behind Alpha's.

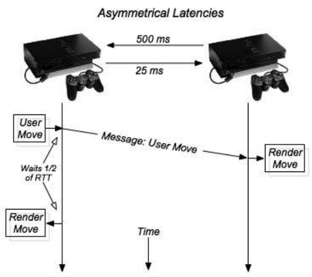

Fig. 7. The first experiment with the inferred latency compensation technique used by online Madden NFL football.

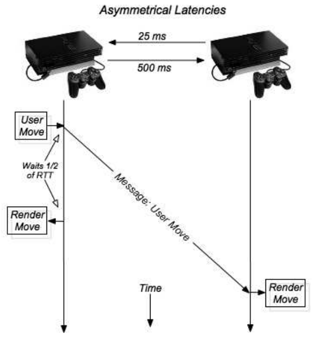

Fig. 8. The second experiment with the inferred latency compensation technique used by online Madden NFL football.

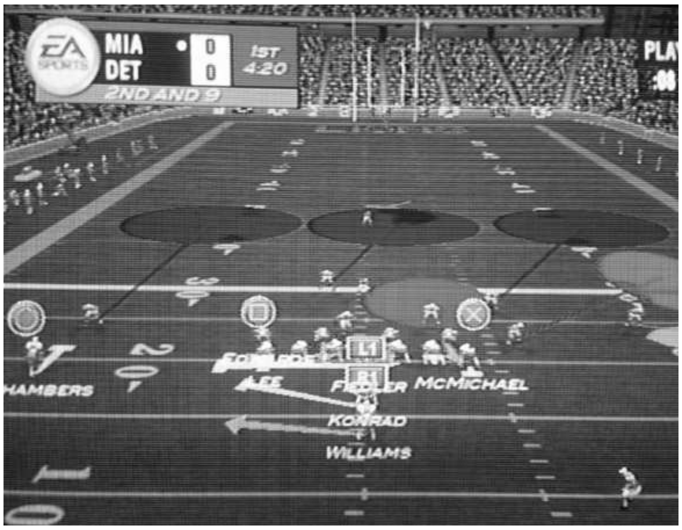

Fig. 9. Offense and Defense plays illustrated.

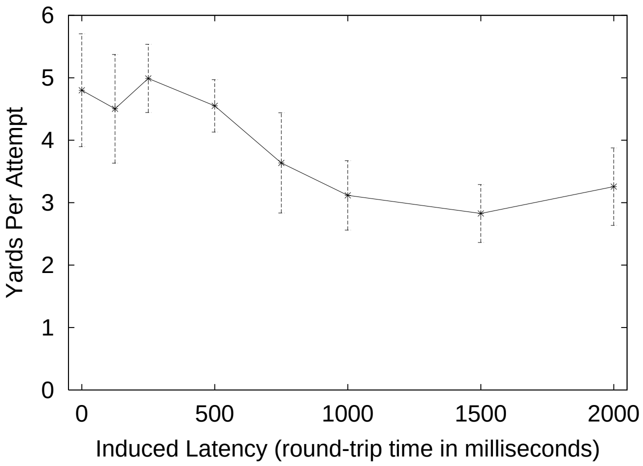

Fig. 10. User Run Performance versus Latency.

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.

IV. Impact of Latency on User Performance

Through pilot studies and hours of play-testing, we propose the primary user interaction components of online football: *running* when a user moves the ball carrier down the field and tries to avoid being tackled; *passing* when a user tries to throw the ball to a receiver down the field; *tackling* when a user tries and get the ball carrier; *coverage* when a user tries to prevent an opponent's receiver from catching the ball; *rushing* when a user attempts to get the quarterback before the ball is thrown; *kicking* when a user kicks the ball to the other team; and *play selection* when a user selects formations and strategies before the play action starts. Because of the nature of football with 11 players on each side and short, complex plays, the user often has little impact on defense once the play starts. Plays that involve kicking are relatively infrequent. Play selection, while important, is similar to turn-based game interaction where each user has seconds to choose their strategy. Thus, we focus on the two most common, fundamental components of offense in this study: running and passing.

We determine ways to quantitatively measure user performance in regards to running and passing. 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 isolate each of the categories of user interactions in Madden NFL football. Studies of other games [SGB+03] enabled construction of custom multi-player maps which isolated each of the user interaction components. Unfortunately, with Madden NFL football there are no "maps" and the game incorporates many non-deterministic components from play to play: receivers run slightly different routes, lineman rush the quarterback differently, 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 any degradation in run performance 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.

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 offense was the Miami Dolphins, a team with a good running back, and the defense was the Oakland Raiders, a team with an average running defense. 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 [JID$^+$04], [LR01] we run more 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 for each average shown with error bars. Over the full range of latencies studied, there is a decrease in performance of about 30%.[10] 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, a round-trip latencies at or below 500 ms is not noticeable to the user. Only after about 750 ms latency or higher is round-trip latency noticeable in that the game feels "laggy." This could explain the relatively flat part of left side of the curve in Figure 10. Anecdotally, if we turn on the NIST Net induced latency during the middle of a play the lagginess is almost immediately perceptible. Second, while playing a game at higher latencies (750 ms or higher) the movements of the player are lagged momentarily behind user input, making it hard to accurately time moves to avoid the defenders.[11] Third, at high round-trip latencies, occasionally a user makes "mistakes" that are unintentional, such as running out of bounds or directly into a defender because the actions of the player are not as fast as the user reactions. We used the instant replay feature[12] 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 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.

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 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. 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.

[10] The correlation coefficient is a pretty strong -0.86, but the relationship between yards per attempt and latency may not be linear based on the visual curvature.

[11] Moves such as spin, juke, stiff-arm, etc.

[12] We recreated a play in a single-player game to take these pictures as the instant replay feature is not available in online Madden NFL.

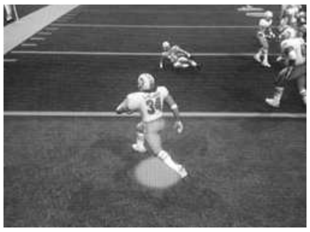

Fig. 11. User is pressing left. Player moves left.

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

Fig. 13. Running back goes out of bounds! User curses.

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

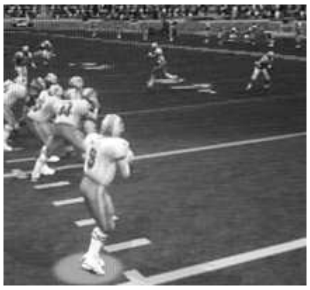

Fig. 15. Throw starts processing here because of the latency.

Fig. 16. Defender intercepts throw! User curses.

Figure 14 depicts the start of play, where, as the receiver begins his route, the user presses the appropriate 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

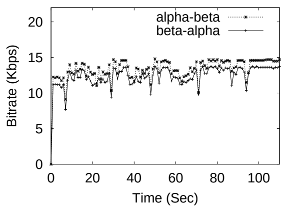

Fig. 17. Bitrate versus Time with no induced latency.

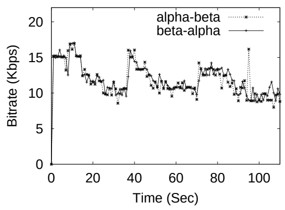

Fig. 18. Bitrate versus Time with 1000 ms induced latency.

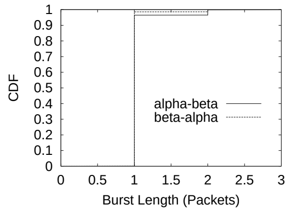

Fig. 19. CDF of Burst Length with no induced latency.

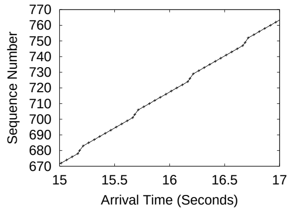

Fig. 20. Packet Sequence Number versus Arrival Time with no induced latency.

the results here due to space constraints. We refer the interested reader to [NC04].

V. 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 games can provide the 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 1000 ms of induced latency, capturing all packets on the Ethernet segment after the NIST Net router. 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 periods that correspond to play selection between play action. The overall average bitrate is low for online Madden NFL football, 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.

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 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. 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. 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.

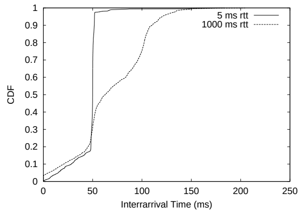

Fig. 21. CDF of Inter-arrival Times.

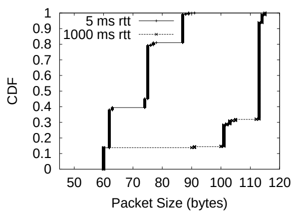

Fig. 22. CDF of Packet Sizes.

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.

VI. 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 500ms being noticeable. However, with latencies higher than 500 ms, running performance can degrade by almost 30%. Overall, we surmize latency artifacts from asymmetric connections are typically dwarfed by the importance of proper play selection; choosing the offensive formation 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 above that of first person shooter games but perhaps below that of real-time strategy games.

Our ongoing work is to determine more effective ways to evaluate latency on passing performance and to more thoroughly explore the effects of latency on running 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 investigation other types of sports games, such as soccer to determine their dependence upon latency, as future work.

References

[Arm01] Grenville Armitage. Lag Over 150 Milliseconds is Unacceptable, May 2001. [Online] http://gja.space4me.com/things/quake3-latency-051701.html.

[Arm03] Grenville Armitage. An Experimental Estimation of Latency Sensitivity in Multiplayer Quake 3. In *11th IEEE International Conference on Networks (ICON)*, September 2003.

[Ber01] Yahn W. Bernier. Latency Compensating Methods in Client/Server In-game Protocol Design and Optimization. In *Proceedings of the Game Developers Conference*, February 2001. [Online] http://www.gdconf.com/archives/2001/bernier.doc.

[BT01] Paul Bettner and Mark Terrano. 1500 Archers on a 28.8: Network Programming in Age of Empires and Beyond. *Gamasutra*, March 2001. [Online] http://www.gamasutra.com/features/20010322/terrano_02.htm.

[egm03] SOCOM: U.S. Navy Seals II. *Electronic Gaming Monthly*, (171):120–125, October 2003.

[Hen01] Tristan Henderson. Latency and User Behaviour on a Multiplayer Game Server. In *Proceedings of the Third International COST Workshop (NGC 2001)*, number 2233 in LNCS, pages 1–13, London, UK, November 2001. Springer-Verlag.

[Int01] International Digital Software Association. Economic Impacts of the Demand for Playing Interactive Entertainment Software, 2001. [Online] http://www.idsa.com/releases/EIS2001.pdf.

[JID+04] Sharad Jaiswal, Gianluca Iannaccone, Christophe Diot, Jim Kurose, and Don Towsley. Inferring TCP Connection Characteristics Through Passive Measurements. In *Proceedings of IEEE Infocom*, April 2004.

[Jus03] Justin Calvert. Record-breaking Season for Madden and NCAA, 2003. Based on EASports Press Release. [Online] http://www.gamespot.com/ps2/sports/madden/-news_6074465.html.

[Lin99] Peter Lincroft. The Internet Sucks: Or, What I Learned Coding X-Wing vs. Tie Fighter. *Gamasutra*, September 1999. [Online] http://www.gamasutra.com/features/19990903/lincroft_01.htm.

[LR01] Dmitri Loguinov and Hayder Radha. Measurement Study of Low-bitrate Internet Video Streaming. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[Mkc00] 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 Measurment and Modeling of IP Traffic*, pages 1 – 11, September 2000. [Online] http://www.caida.org/-outreach/papers/2000/AIX0005/AIX0005.html.

[NC04] James Nichols and Mark Claypool. The Effects of Latency on Online Madden NFL Football. Technical Report WPI-CS-TR-04-06, CS Department, Worcester Polytechnic Institute, March 2004.

[Ng97] Yu-Shen Ng. Designing Fast-Action Games for the Internet. *Gamasutra*, September 1997. [Online] http://www.gamasutra.com/features/19970905/ng_01.htm.

[PW02] Lothar Pantel and Lars C. Wolf. On the Impact of Delay on Real-Time Multiplayer Games. In *Proceedings of the Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, May 2002.

[SERZ02] Christian Schaefer, Thomas Enderes, Hartmut Ritter, and Martina Zitterbart. Subjective Quality Assessment for Multiplayer Real-Time Games. In *Workshop on Network and System Support for Games*, April 2002.

[SGB+03] Nathan Sheldon, Eric Girard, Seth Borg, Mark Claypool, and Emmanuel Agu. The Effect of Latency on User Performance in Warcraft III. In *Proceedings of the ACM NetGames Workshop*, May 2003.

[SKH02] J. Smed, T. Kaukoranta, and H. Hakonen. Aspects of Networking in Multiplayer Computer Games. *The Electronic Library*, 20(2):87–97, 2002.

[The03] The Entertainment Software Association. Essential Facts About the Computer and Video Game Industry, 2003. Press Release. [Online] http://www.theesa.com/EF2003.pdf.

Appendix

This appendix contains text and graphics that we could not include in the regular paper body because of space constraints.

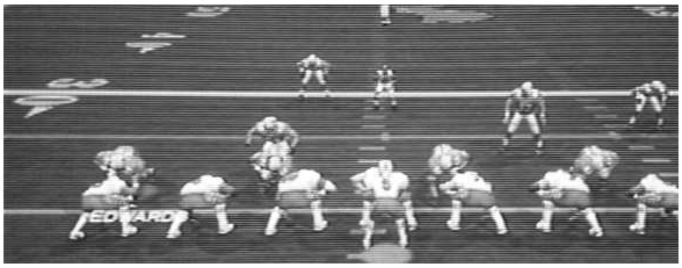

Fig. 23. Defensive Formation.

Additional Statistics

List of possible statistics to use to application-level performance:

- Offense
  - Yards run/passed for
  - Run/pass attempts
  - Pass completions
  - Broken tackles
  - Longest run/pass
- Defense
  - Tackles
  - Deflected passes
  - Intercepted passes
  - Quarterback sacks/knockdowns
- Kicking
  - Yards per kick
  - Punts inside 20 yard line

Additional Screen-Shot of Defense

Figure 23 illustrates the right-side positioning of the defensive formation.

Additional Pilot Studies

The offense ran, then punted immediately to the defense. For early pilot studies, we ran this combination of offensive and defensive plays repeatedly for the entire length of a game with ten minute quarters.[13] We played the 1st quarter with NIST Net disabled, at the fundamental latency of the testbed. At the start of the 2nd quarter we set an induced, symmetrical round-trip time of 2000 ms at the NIST Net router. We repeated this procedure for the second half of the game, with no induced latency in the 3rd quarter, but a 2000 ms round-trip time in the 4th. We alternated like this to see if there are any effects of the players getting tired. We averaged the results for each of the quarters played at the same latency and compare these numbers to the average obtained from playing the whole game at the same latency. We found the results to be similar, any slight differences can be attributed to the inherent statistical variance in the game. However, we are uncertain as to what effects the change in latency from quarter to quarter has on the user, so for subsequent experiments we fixed the latency and play an entire game at that latency.

[13] While real football typically has 15 minute quarters, 10 minute quarters is typical for online play.

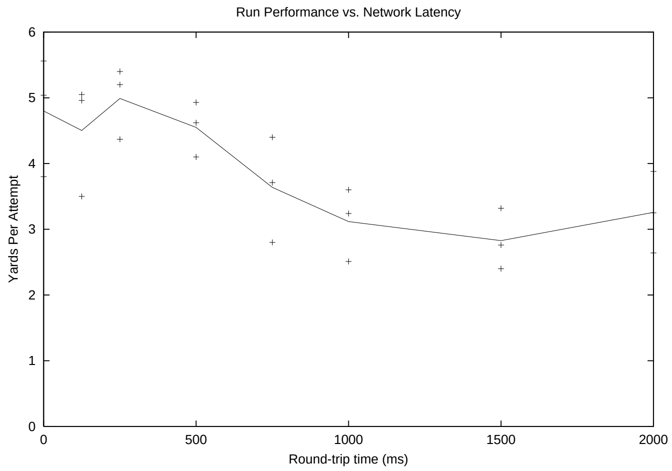

Fig. 24. User Run Performance vs. Latency

Additional Run Data

Figure 24 shows the three samples of the average yards per attempt for each game versus the latency, along with the line connecting the average of the three samples.

Additional Pass Data

To better quantify the timing aspects of passing we ran some additional experiments. We ran a passing play where the receiver runs straight down the field. We moved the quarterback to be directly in line with the receiver and threw the ball. Next, using the instant replay feature and a stopwatch we measured how long it takes for the ball to get from the quarterback to the receiver. Then we measured how many yards the ball traveled and divided the distance by the time to get the approximate football speed. We find it takes about 0.6 seconds for the ball to travel 20 yards, so the travels at about 70 miles per hour (33 yards/sec). This result is confirmed by anecdotal evidence found on the Internet and heard during real NFL games that some professional quarterback can throw the ball 60 miles per hour.

Our second experiment was to determine the approximate speed of the players in the game. We measured this by running a “practice” with the same team as before. We executed a play and threw the ball to the receiver that ran the route we discussed in the passing play shown in Figures 14-15. Then we ran this receiver straight up the field, feasible since during a practice play and there are no defensive players. We measured how long it took for the player to run 50 yards, which is about 5.8 seconds, for a speed of 17 miles per hour (8 yards/sec). This is very fast, equivalent to a sub-4 minute mile. Next, we ran the same experiment but hold the “sprint” button down, and the receiver ran about 20 miles per hour (10 yards/sec). Perhaps these super-human running speeds were implemented to make players appear more athletic.

Finally, we measured how long it takes from the start of the play shown in Figures 14-15 to when the receiver is “open.” We find that it takes about 1 second. Therefore, for the play to be successful, the user must see that the defense is in zone coverage, and press the button for the pass about 1/2 a second after the play starts. This quickness is necessary because it takes about another 1/2 seconds for the ball to travel through the air, and it takes about 1 second for the receiver to get open. Otherwise, any defenders within 4-5 yards of the receiver, as in our example above, will be able to close the distance between them and the receiver and knock down or intercept the pass. If there is latency on the network above 500 ms, it makes it nearly impossible to complete the pass successfully. Thus, latency also has an effect on the user performance of passing.