Draft: document last updated: December 11, 2011

Assessing the Impact of Loss and Latency on the Perceived Quality of a Modern First Person Shooter Game

James Martin, Ph. D.
Associate Professor
School of Computing
Clemson University
jim.martin@cs.clemson.edu
864.656.4529

Juan E. Gilbert
Chair Human-Centered Computing Division
Professor
School of Computing
Clemson University
juan@clemson.edu
864.656.4846

Students:
Rahul Amin (ramin@clemson.edu), France Jackson (fljacks@clemson.edu), Marvin Andujar (andujarm@kean.edu), Marquita Frazier (marqutf@g.clemson.edu), Andrea Johnson (Andrea5@clemson.edu), Pharen Johnson (JohnsonPM@paine.edu), Philip Hall (pwhall@g.clemson.edu)

Introduction

Resource allocation is the process by which network elements try to meet the competing demands that applications have for network resources. While broadband cable networks have steadily increased the size of the 'pipe' available to subscribers, end user demand for bandwidth is insatiable. Therefore, network congestion is a fact of life. The response by each application to congestion differs widely. File transfer applications such as FTP or P2P file sharing are typically the most flexible in performance requirements. Applications that involve real-time end user interaction, such as web browsing, online shopping, or online games are less flexible as they generally require low latency. Emerging web-based video streaming are generally flexible; however, once performance thresholds are surpassed, perceived quality drops off quickly. Finally, over-the-top voice and high quality video broadcasts are the most sensitive to fluctuation in bandwidth, latency and loss.

Our research focuses on network gaming applications. In particular, we study real-time online games. The academic community has studied the impact of latency and loss on real-time games such as shooter or action games [BC04, QM03, SG03]. The wide range of previous results confirms that mapping network performance metrics such as latency and loss to an assessment of perceived quality is extremely difficult. The challenge is in part caused by the fact that each game responds to latency and loss is unique ways. Further, for a given game, specific modes of operations and even game play scenarios (which change quite frequently) will react differently to latency and loss. Perhaps the most problematic aspect of assessing user's quality of experience is due to the fact that a subjective assessment is a measure of how the gaming experience fails to meet a user's expectation. Accurately quantifying expectations is very difficult as a gamer's expectation depends on past history and personnel factors.

Works such as [BC04,SG03,QM04,WVW10] all seek to understand the impact that loss and latency have on real-time online games. The usual approach is to emulate loss and latency in a controlled testbed environment. In [BC04] the authors found that game play was viewed as less enjoyable at latencies over 100 ms. In [QM04] players thought they detected impairment once delays exceeded 60 ms. More recent work has tried to find commonality of results over classes of games [CC06,CC10].

Our research is motivated by the observation that much of the prior work described in the literature has focused on objective studies. While subjective studies have been considered, there have been limited results. This is primarily because a widely accepted utility function that maps objective metric results to a perceived quality assessment has not been established. We focus on the widely played FPS game, Call of Duty Modern Warfare II. We engage the student population at Clemson University to participate in a large scale experimental study designed to explore the impacts of network impairments on the perceived quality. We collect information from the participants including their self-assessed gaming expertise, scores achieved during the testing session, and assessment of perceived quality. While the research described in this paper is primarily exploratory in nature, we did focus our efforts by considering the following hypothesis:

the perceived quality correlates to a gamer's skill level. If this were true, we would expect to see different thresholds of tolerance of network impairment based on the level of experience and skill. We expect that the methodology developed in this study will pave the way for the following intriguing questions to be considered in future work:

- Is it possible to calibrate a gamer's expectations through a calibration phase, and therefore possibly enhance the perceived quality of the gaming experience?
- Is a gamer's assessment of perceived quality influenced more by inconsistent service levels rather than consistent (but lower) levels of service?
- Are there human or social behavior factors that can be taken advantage of through game design and interface techniques or through network services that could reliably compensate for network impairment?

This paper is organized as follows. The next section surveys the literature and summarizes relevant related work. This is followed by a description of our experimental methodology. The next section documents our results and analysis. Finally, we provide conclusions including a discussion of the limitations of our study and possible next steps.

Background

Our study focuses on multi-player, online games. While this broad classification includes non-real-time games such as board games or fantasy games, many involve real-time interactions. The work in [CC06] presents a useful taxonomy for classifying real-time online games. The authors correlate the game's relationship between latency requirements and 'precision' (either movement or shooting) requirements. Further attributes include the interaction model and the game perspective. The interaction defines how a player interacts with the game world. In [CC06], the authors indicate the interaction is either through an avatar or with an omnipresent model. The avatar model exists at a particular location in the game context and is only involved with local interaction. The omnipresent model provides players a global view of the game. The game perspective is tied to the interaction model as it defines how players view the game world. For avatar interaction, such as in shooter games, a first or third person shooter perspective is common. For omnipresent interactions, such as in role playing games, the perspective might be variable. A player might have access to aerial views as well as first or third person views.

Because real-time online games are sensitive to network and system performance, there has been a significant amount of prior academic research in this area. Many of these studies share our goals and methods. Works such as [BC04,SG03,QM04,WVW10] all seek to understand the impact that loss and latency have on real-time online games. The usual approach is to emulate loss and latency in a controlled test-bed environment. Either objective measures such as game results or subjective tests based on Mean Opinion Scores (MOS) are used. The impacts of loss are arguably not as significant as that caused by latency because most games are designed to tolerate some amount of loss either by sending game state or control commands frequently or by utilizing error correction. Therefore, much of the research presented in the literature has focused on the impacts of latency.

In [BC04], the authors found that the ability to carry out ‘precise’ movement and shooting tasks is not significantly affected by loss rates or latency that are typical for the FPS game Unreal. They did observe players scores (in terms of kills and deaths) in some game modes drop as the latency increased beyond 100 milliseconds (ms). In [SG03], the authors focus on the effects of latency on World of Warcraft. They find that because the game is real time strategy (rather than first person shooter), high levels of induced latency (well beyond 100 ms) is noticeable but has negligible effect on the outcome. Other works have used methods involving subjective tests where gamers were asked about their perceived quality. In [LA03], as round trip times exceed 150 ms, the performance of Halo gamers decreased by about 50% while the perceived quality was still high. Further, games with over 500 ms of delay and 4% loss were considered unplayable. The work in [QM04] shows that in some scenarios (i.e., play situations of a specific FPS game), even small amounts of latency and jitter (as small as 60 ms) caused users to notice a drop in perceived quality.

The work examined in [CC06] attempted to group the games in prior studies based on model (avatar or omnipresent), perspective (first or third person or varied), and genre (FPS, Racing, Sports, RPT, RTS) to one of three levels of latency sensitivity. The thresholds of induced latency for high, medium and low sensitivity settings were estimated to be 100 ms, 500 ms, and 1,000 ms.

More recent work [CC10] has addressed this by exploring the effects of latency on classes of games. Using the open source third-person shooter game BZFlag, the authors illustrate that the range of tolerated latency can range by an order of magnitude (from less than 100 ms to 1.0 seconds) depending on the required precision for game play.

The work in [CZ11] provides a useful framework for describing and assessing game quality of experience through the use of ‘perceptual view inconsistency’. The framework assumes that 1) Quality of Experience (QoE) is based on an overall assessment that is based on individual (and decomposable) perceptions; 2) Inconsistencies result from the disconnect between the gamers view and the server’s (or global) view; 3) An overall QoE assessment requires each objective inconsistency to be mapped to corresponding subjective values. The authors refer to prior measurement work and show that the degradation of various metrics of perceived quality, each of which are specific to certain segments of specific games, can be modeled as a half normal curve.

Based on the wide range of results that exists in the related work, we draw the following motivating conclusions:

- The impacts of latency and loss depend not just on the class of game, but also on particular modes of play and on specific gaming contexts.
- Establishing the relationship between latency and loss network measures with perceived quality is the most challenging issue that remains largely unsolved.

Our study develops a methodology for exploring the complex interaction between network impairment, the impacts on FPS online games, and the quality of experience perceived by end users (i.e., the gamers).

Methodology

Figure 1 illustrates the experimental setup. All results described in this paper are based on experiments involving Call of Duty Modern Warfare II running on an Xbox 360 gaming system. Our testing involves a set of seven players in a private game (i.e., unknown players are not allowed to join the game). The game mode is 'free for all' with a time limit of 10 minutes. There is no score limit (i.e., 'free for all' supports a mode where the game is over once the first player reaches the game score limit).

As illustrated in Figure 1, five control gamers are used to establish a consistent level of play for all tests. The control players are Clemson students who were paid for their participation. The control players are all experienced Call of Duty players.

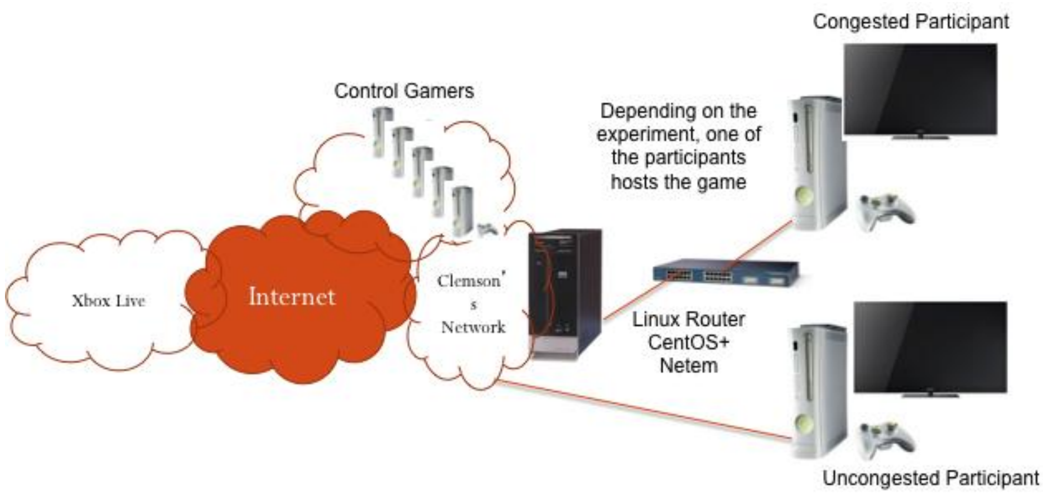


Figure 1. Experiment Overview

During a testing session, two additional students play the game. These are the gamers under observation and are referred to as the participants. One participant uses an Xbox that connects to the Internet through a controlled subnetwork. A Linux router is configured to apply a specific network impairment on all traffic sent and received using the kernel's network emulation (netem) capability[1]. We refer to this player as the congested participant. The uncongested participant is directly connected to the Internet (through the Clemson network). During a testing session, the participants in the lab do not see each other's LCD screens. The participants are not given a microphone; consequently, they do not use the online chat capability of the game. The five control gamers however are allowed to converse if they desire. The participants hear all sounds

[1] Please refer to this site for further information on netem :
http://www.linuxfoundation.org/collaborate/workgroups/networking/netem

generated by the Xbox as well as all discussion between the control gamers. The five control gamers are physically located either on or off campus. In the former case, the control gamers are located in student housing with direct connectivity to the campus network. In the latter case, they connect through an Internet Service Provider to reach the Internet.

Each participant volunteers 30 minutes of his/her time to participate in at most one testing session. On arrival to the lab, they are given instructions and then asked to fill out a pre-survey questionnaire (refer to Appendix A). The group plays for 10 minutes. Upon completion all players' (including the control gamers) results are recorded. A player's results include the number of times they died (referred to as deaths) and the number of kills achieved by the player. The participants are then asked to fill out a post-survey questionnaire (refer to Appendix B). Based on these responses, we compute a Mean Opinion Score (MOS). The computation is based on several pieces of information provided in the post-survey questionnaire. Refer to Appendix C for the algorithm that computes the MOS score.

Table 1. Network Impairment Settings

Setting Identifier	Game Host	LatencyTarget (ms)	Jitter (ms)
1	Uncongested Participant	100	0
2	Uncongested Participant	100	100
3	Uncongested Participant	100	250
4	Uncongested Participant	100	300
5	Congested Participant	100	0
6	Congested Participant	100	50
7	Congested Participant	100	100
8	Congested Participant	100	150

For each session, a network impairment setting is selected. There are eight possible settings as shown in Table 1. Each setting is defined by the location of the host (i.e., which of the participants is hosting the game) and the desired amount of latency and jitter. When hosting a game, the host Xbox performs a server function in addition to operating as a client. All players' movement and command data are sent to the server which then distributes to all clients in the game. As mentioned, we rely on the Linux netem capability to provide the additional latency and jitter. For the results discussed in this paper, we do not add artificial packet loss. All inbound packets (from the campus network to the congested participant) and outbound packets (from the congested participant to the Internet) are artificially delayed on average by 100 milliseconds. The set of latency/jitter settings are slightly different depending on which participant hosts the game. This is because when the congested user is the host, all other gamers connect to this host and so streams from multiple users will compete for bandwidth on a congested network. Whereas when the non-congested user is the host, the only traffic on the congested network is the packet stream sent between the congested user and the game host. This results in an increased amount of traffic on the congested network for settings 5 through 8 as compared to setting 1 through 4. Consequently, the jitter values for settings 5 through 8 are lower than those used for settings 1 through 4 as seen in Table 1.

The next section describes in more detail the effects of the different netem settings that are used in the experiments.

Calibration Experiments

We begin by overviewing the network consumption and dynamics associated with the gaming test sessions. When the game host is located at the non-congested participant, the traffic sent and received by the congested particpant's Xbox is dominated by a single bi-directional low bandwidth UDP flow between the particpant's Xboxes. We obtain a tcpdump trace of one of the test sessions that utilizes network emulation setting #2 (with a latency and jitter setting of 100 ms and 100 ms respectively). Figure 2 illustrates where the stream is captured with respect to where the netem latency is applied. For outbound packets (i.e., packets generated by the private network), the tcpdump captures the stream before netem has applied the additional latency. For inbound packets, the tcpdump captures the arrival stream after the netem module.

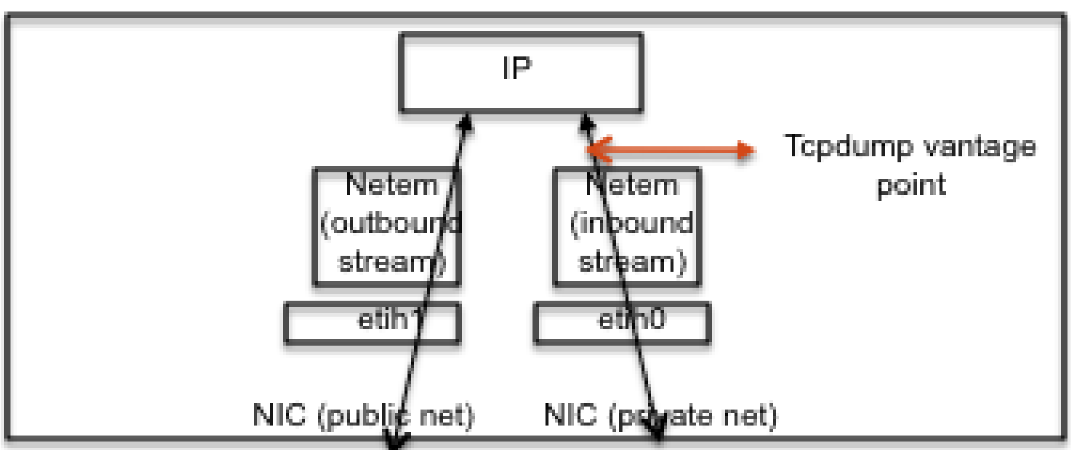


Figure 2. Linux Structure

Table 2 lists all flows that were observed in the trace. Our tool samples bandwidth using a timescale of 1 second. The average throughput column in Table 2 is the sample mean. The minimum and maximum throughput identify the minimum and maximum sample observed in any 1 second interval. The last column identifies the standard deviation of the samples in seconds. The IP addresses correspond to the gamers in the following manner:

- 10.109.12.247 – Congested User
- 130.127.49.224 – Non-congested User (Game Host)
- 66.112.166.187 – Gamer 1
- 130.127.98.31 – Gamer 2
- 108.192.150.113 – Gamer 3
- 130.127.92.244 – Gamer 4
- 130.127.72.250 – Gamer 5

Figure 3 visualizes the flows observed in the trace. Outbound packets over the dominant flow consume an average bandwidth of 25.6Kbps. Packets are sent at roughly fixed intervals with an observed mean of 36.63 ms and a standard deviation of 6.7 ms. The average packet size is 126 bytes. Figure 4 illustrates a snapshot of packet interarrival times for the inbound and outbound traffic. The mean interarrival time for inbound packets is 100.14 ms and the standard deviation is 71.88. The inbound stream consumes roughly half as much bandwidth (average of 12.75 Kbps) as that of outbound stream (average of 25.60 kbps). We conjecture that the client generates periodic information updating the server with action events (movement, gun selection, shooting commands). The server periodically updates clients with client-specific information as well as global information at a reduced rate or in a more efficient manner. Figure 3 shows a very small amount of traffic between the congested gamer and the other gamers. We conjecture that this traffic is related to conversations engaged in by the 5 control gamers..

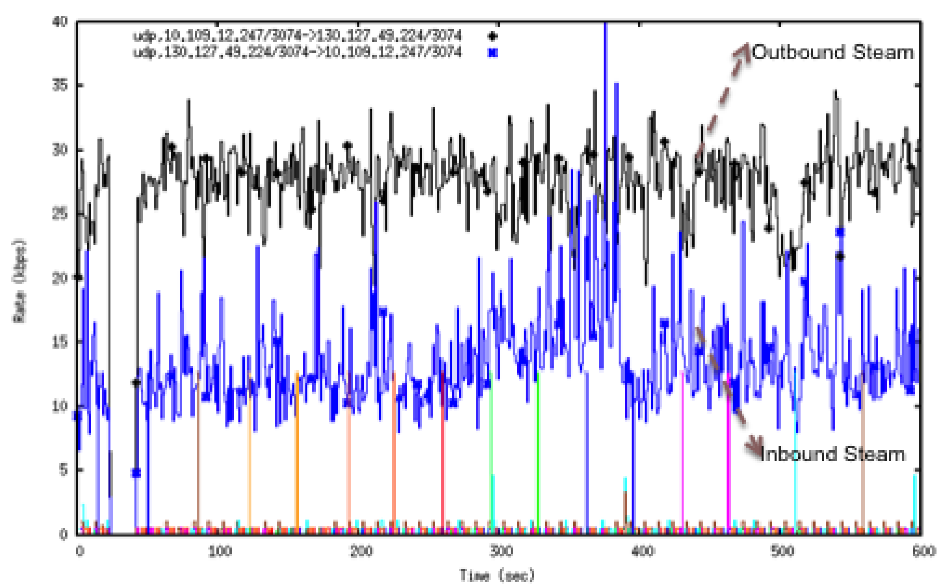


Figure 3. Bandwidth Consumed in an Experiment Involving Network Impairment #2

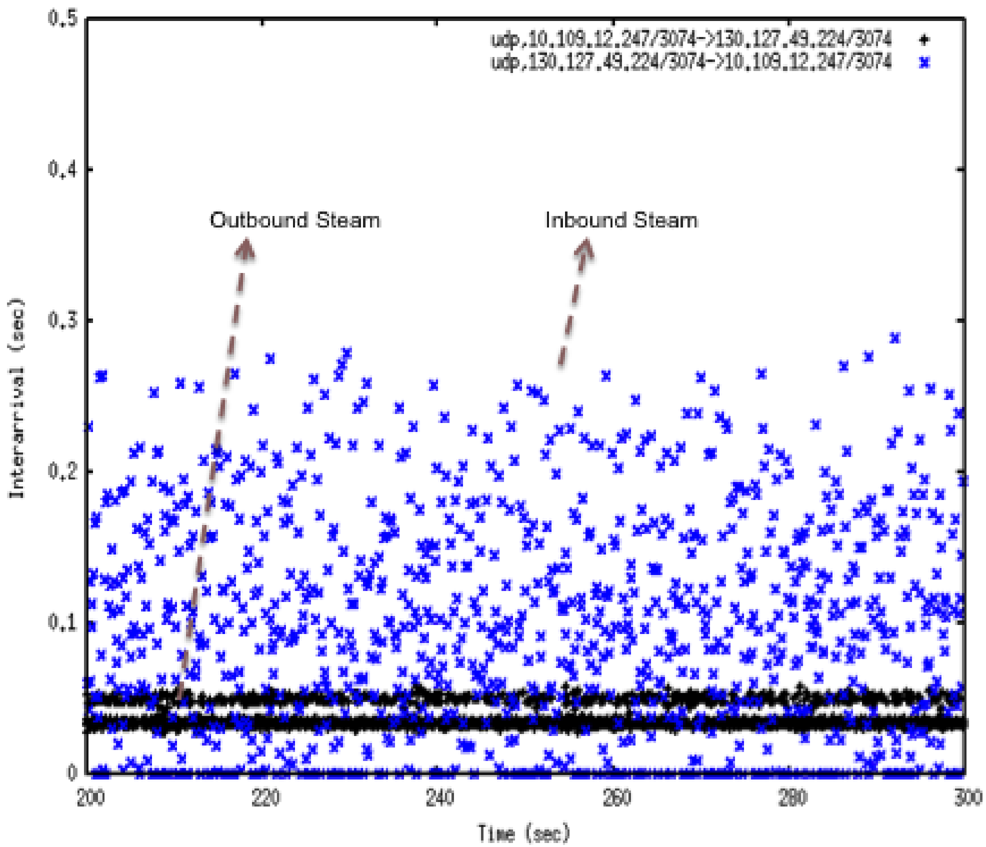


Figure 4. Packet Interarrival Times for Experiment Using Network Impairment #2

We also examine the bandwidth consumption and dynamics when the congested participant's Xbox serves as the game host. In this case, we observe 6 bidirectional UDP flows (between the server and the 6 other game clients). The dynamics of each UDP flow is very similar to that observed in the case when the uncongested participant serves as the host (i.e., each of the 6 flow's bandwidth consumption is virtually identical to that shown in Figure 3).

Table 2. Active flows during a testing session

From	To	Mean Bandwidth (kbps)	Minimum Bandwidth (kbps)	Maximum Bandwidth (kbps)	Standard Deviation (seconds)
10.109.12.247	130.127.49.224	25.5977	0	34.656	7.3223
130.127.49.224	10.109.12.247	12.754	0	41.60	5.01
10.109.12.247	66.112.166.187	0.101	0	1.024	0.234
66.112.166.187	10.109.12.247	0.100	0	1.024	0.295
10.109.12.247	130.127.98.31	0.099	0	0.544	0.203
130.127.98.31	10.109.12.247	0.099	0	0.544	0.202
10.109.12.247	108.192.150.113	0.099	0	0.544	0.203
108.192.150.113	10.109.12.247	0.099	0	0.544	0.202
10.109.12.247	130.127.92.244	0.099	0	0.544	0.203
130.127.92.244	10.109.12.247	0.099	0	0.544	0.203
10.109.12.247	130.127.72.250	0.099	0	0.544	0.203
130.127.72.250	10.109.12.247	0.099	0	0.544	0.203

To assess and validate the netem impairment settings we developed a client-server performance tool to help us assess network performance that might be experienced by a live Xbox session. The client, which is located on a Linux box connected to the Clemson network, sends a stream of UDP packets to a second Linux box that is connected on the private network. The tool sends fixed size packets of 100 bytes at a constant rate of 30 Kbps (i.e., the tool loosely mimics the traffic generated by an Xbox client). Table 3 summarizes the observed effects of the netem impairment settings. These results are the average of six runs of each setting. The run time is 10 minutes which is the duration of the user tests. The 99% confidence interval for the observed latency, jitter, and standard deviation is within 1% of the statistic.

Table 3. Assessing the netem Jitter Setting

Netem Setting (Latency, Jitter in ms)	Avg. Latency (ms)	Latency Standard Deviation (ms)	Jitter (ms)	Out of Order Packets (Percenta
100, 0	100.127	0.0275	0.176	0
100,50	100.81	28.895	33.2631	0.3166
100,100	104.4	54.328	61.0025	0.5248
100,150	116.125	71.839	79.7147	0.577
100,200	129.35	88.766	99.2527	0.6038
100,250	140.785	103.984	116.677	0.611
100,300	153.391	120.432	131.803	0.61

As noted in [AS04,JL11], netem's latency module has flaws and consequently requires explanation. When netem is configured to generate uniform random jitter, it accepts two parameters which we call *LatencyTarget* and *Jitter*. The *LatencyTarget* is the desired average per packet latency. The *Jitter* is the maximum positive range of the amount of per packet jitter that could be added to a constant latency that is applied to each packet. The actual per packet delay is:

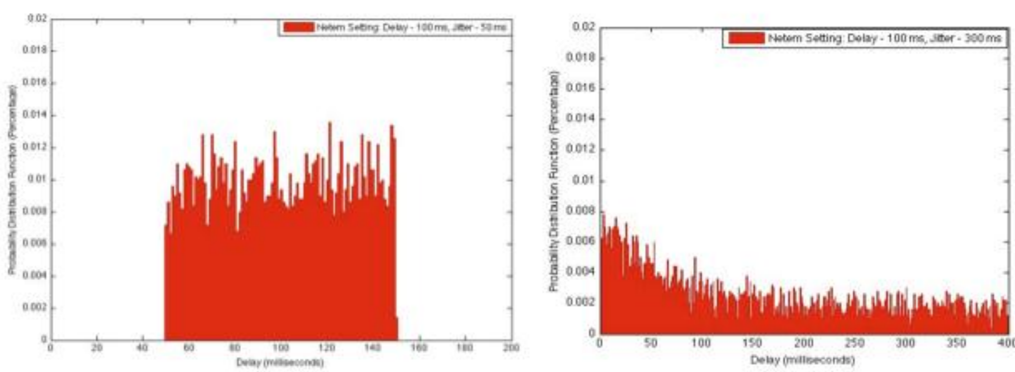
Packet Delay = LatencyTarget + Uniform[-Jitter, +Jitter]

The maximum per packet latency that should ever be applied to a packet is:

maxLatency =LatencyAverage+Jitter;

As long as the *Jitter* is less than the *LatencyAverage*, the set of delay values will indeed be uniform in the range *[LatencyAverage-Jitter*, *maxLatency*]. However, it is possible for packets to be delivered out-of-order. For example, if the i'th packet is assigned a delay value of 150 ms and the (i+1)'th packet is assigned a delay value is 5 ms, the second packet will be sent before the first.

When the *Jitter* exceeds the *LatencyAverage*, netem will overwrite the negative delay value in a manner that skews the distribution of delay samples. Figure 5a plots the average latency experienced by packets between the two test machines (i.e., in the inbound direction) when the *LatencyAverage* is 100 ms and the Jitter is 50ms. The measured average delay is 100.81 ms and the measured standard deviation is 28.90 which are both expected from a uniform distribution with a range of [50,150] ms. Figure 5b illustrates the skew that occurs for a case where the Jitter exceeds the *LatencyAverage*. In this latter case, the LatencyAverage is 100 ms and the Jitter is 300 ms. The resulting mean observed latency is 153.39 ms and the standard deviation is 120.43 ms. If the netem algorithm simply throws out samples that are less than 0 ms and computes a new sample, we would expect a uniform distribution with a mean observed latency of *LatencyAverage*. The actual algorithm used by netem is not known.



a. Added Packet Delay with a Jitter of 50 ms

b. Added Packet Delay with a Jitter of 300 ms

Figure 5. Average Packet Delay Test Using Jitter of 50 ms and 300 ms

Table 2 shows a significant level of out-of-order packets. However, once the Jitter reaches 200 ms we see that the rate of out-of-order packets converges to about 0.60. The tool considers any packet that arrives with a sequence number less than the next expected is out-of-order. Therefore, if packet sequence numbers 1,10,2,3,4,5,6,7,8,9 arrive, 8 are considered out-of-order (packet sequence numbers 2 through 9). So the out-of-order statistic shown in Table 3 is slightly misleading. Figure 6 visualizes the arrival pattern of packets sent by the performance tool when netem is set to a Jitter value of 300 ms. For this jitter setting, the figure plots the sequence number and time of arrival of the arriving packets. We assume that the game does some amount of packet reordering and possibly loss recovery through error correction. However it is reasonable to assume that the impairment experienced by the game caused by the netem setting is a combination of increased packet delay and packet loss. ***The advantage of this method is that it provides a wide range of operating conditions for the assessment. The disadvantage is that it makes it difficult to know how to map specific network measures (loss and latency) to QoE estimates.***

The delays in the game that are observed when the netem setting (latency/jitter) is (100ms, 100 ms) is roughly similar to when the latency is set to a constant 200 ms (and the jitter is 0 ms). Similarly, the delays in the game that are observed when the netem setting (latency/jitter) is (100ms, 300 ms) is roughly similar to when the latency is set to a constant 450 ms. However, when netem is set to a fixed latency (with no jitter) the perceived experience is different. The students participating on the project indicated the game impairment is predictable when the jitter is set to 0 ms. The students also agreed that the impairment is more realistic and hard to adjust to when random jitter is used. For example, when the congested user tries to throw a grenade in the constant latency and no jitter case, he/she can adjust to the impairment setting by making a mental note of approximately how much quicker he/she should launch that grenade to attain the desired result. Trying to approximate this phenomenon in the case when there is jitter is much more difficult and frustrating to the congested user.

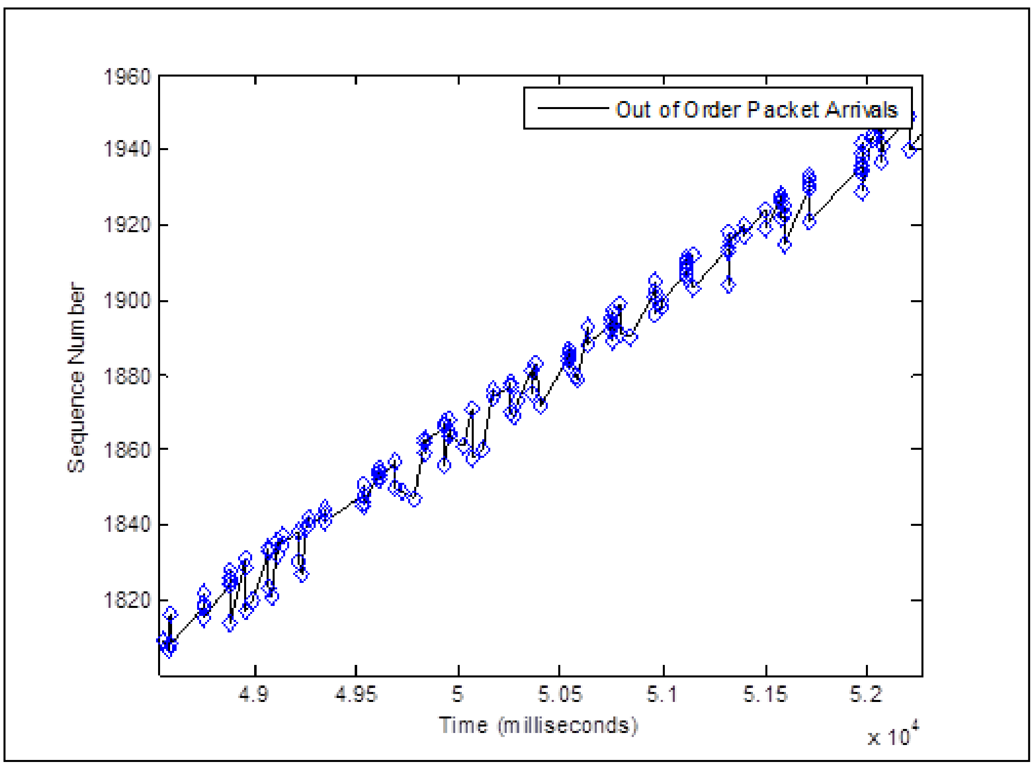


Figure 6. Arrival of Packets from the Performance Tool (Latency/Jitter Setting of 100ms/300ms)

Experiment Formulation

For each of the eight scenarios, we will perform N assessments. The dependent variables in the study are the value of the jitter and the location of the Host game server. Two statistical tests are needed to determine an appropriate value of N. First, since the test methodology involves randomness (i.e., the netem jitter process), we need to ensure that the game session time and the number of samples is sufficient to allow the statistics to converge. Second, we need to have a sufficient N so that the statistical accuracy of the survey is within a desired bound.

[Need to finish writing up this analysis…the point to make is that the statistical accuracy of our results depends on the data. In other words, we will indicate observed results (based on 64 test sessions) and provide the statistical accuracy]

Results and Analysis

This section will be written once all data has been acquired. Appendix 3 summarizes one Mean Opinion Score that we developed. Figures A1 and A2 visualize results based on a very small number of samples. The figures have been added just to help explain and illustrate the experimental methodology.

Conclusions

Touch on:

- This is what we did
- Our key contribution is …
- We had the following restrictions…..

Future work:

- Compare the results with a 2nd game (like World of Warcraft)
- Include the quality of audio in the 'experience'
- Change method where each user is exposed to a baseline
- Time varying impacts (impairment after a certain amount of time perceived more than early)

References

[AS04] G. Armitage, L. Stewart, "Some Thoughts on Emulating Jitter for User Experience Trials", Proceedings of SIGCOMM'04, August 2004.

[BC04] T, Beigbeder, R, Coughlan, C. Lusher, J. Plunkett, E. Agu, M. Claypool, "The Effects of Loss and Latency on User Performance in Unreal Tournament 2003", Proceedings of 3rd ACM SIGCOMM Workshop on Network and System Support for Games (NetGames-04), August, 2004.

[CC06] M. Claypool, K. Claypool, "Latency and Player Actions in Online Games", ACM Communications of the ACM, November 2006.

[CC10] M. Claypool, K. Claypool, "Latency Can Kill: Precision and Deadline in Online Games", Proceedings of Multimedia Systems Conference (MMSys'10), February, 2010.

[CZ11] P. Chan, M. Zarki, "Perceptual View Inconsistency: An Objective Evaluation Framework for Online Game Quality of Experience (QoE)", ACM NetGames '11 Workshop, October, 2011.

[DWW05] M. Dick, O. Wellnitz, L. Wolf, "Analysis of Factors Affecting Players' Performance and Perception in Multiplayer Games", Proceedings of NetGames '05, October 2005.

[JL11] A. Jurgelionis, J Laulajainen, M. Hirvonen, A. Wang, "An Empirical Study of NetEm Network Emulation Functionalities", Proceedings of the 5th International Workshop on Performance Modeling and Evaluation of Computer and Telecommunications (PMECT 2011), August 2011.

[LA03] T. Lang, G. Armitage, "A Ns2 Model for the Xbox System Link Game Halo", Proceedings of the Australian Telecommunications Networks and Applications Conference, 2003.

[QM04] P. Quax, P. Monsieurs, W. Lamotte, D. Vleeschauwer, N. Degrande, "Objective and Subjective Evaluation of the Influence of Small Amounts of Delay and Jitter on a Recent First Person Shooter Game", Proceedings of the SIGCOMM 2004, August, 2004.

[SG03] N. Sheldon, E. Girard, S. Borg, M. Claypool, E. Agu, "The Effect of Latency on User Performance in Warcraft III", In *Proceedings of ACM Network and System Support for Games Workshop (NetGames)*, Redwood City, California, USA, May 2003.

[WVW10] R. Woundy, Y. Lee, A. Veiga, C. Williams, "Congestion Sensitivity of Real-time Residential Internet Applications", Proceedings of the SCTE Cable-Tec Technical Workshop, November 2010.

Appendix 1 Pre-Survey Questionnaire

Gaming Study Pre Survey

1. **Participant Number: _________________________**

2. **Age: ___________________________**

3. **Gender: Male or Female circle one**

4. **How do you describe yourself?**
   American Indian or Alaska Native
   Hawaiian or other Pacific Islander
   Asian or Asian American
   Black or African American
   Hispanic or Latino
   Non-Hispanic White

5. **How many hours a week do you play?**
   (0-20) (21-40) (41-60) (61-80) (81+)

6. **What types of games do you play?**
   MMOG Sports Racing Shooter Other
   If other,____

7. **Name the top 3 games that you play?**

8. **What consoles do you play with?**
   Xbox360 PS3 Wii PSP Nintendo DS PC Other
   If other____

9. **Which console do you play with the most?**

10. **Do you use an Ethernet (wired) or wireless connection**

11. **Who is your service provider? (if you live on campus, please answer based on your parents broadband service)**
    a. **What is the speed of your connection?**

Appendix 2 Post-Survey Questionnaire

Gaming Study Post Survey

1. **How satisfied were you with your gameplay?**

   Very Satisfied   Satisfied   Neutral   Dissatisfied   Very Dissatisfied

2. **If you were dissatisfied, please explain why?**

3. **Were you frustrated you during your gameplay?**

   Yes or No

4. **If yes, rate your level of frustration**

   Very Frustrated   Frustrated   Neutral   Somewhat Frustrated   Not Frustrated

Oral Questions:

1. **Based on your gameplay, how much do you think the lag impacted your gameplay experience?**

   A lot   Quite a Bit   Somewhat   Very Little   No Impact

2. **Based on your observed network conditions during gameplay, how likely are you to change your service providers?**

   Very Likely   Somewhat Likely   Neutral   Unlikely   Very Unlikely

Appendix 3 MOS Calculation

The pre-survey and post-survey answers obtained via the participants after each game experiment session are used in coming up with a Mean Opinion Score for each network impairment setting. The pre-survey responses are used to come up with a numeric mapping between 1-5 for each participant's experience level as shown in Table A1. The post survey responses are similarly mapped to a numeric value between 0-4 as shown in Table A2-A5 to determine each user's gameplay satisfaction, gameplay frustration, perceived impact of lag on gameplay and likeliness to change service provider respectively. Giving each of the post-survey numeric mappings equal importance, an opinion score value for each participant is computed according to the following formula:

Opinion Score per Participant = (0.25 * Gameplay Satisfaction) + (0.25 * Gameplay Frustration) + (0.25 * Impact of Lag on Gameplay) + (0.25 * Likeliness to Change Service Provider)

To come up with the final mean opinion score, the average of each participant's opinion score is used as follows:

$$\text{Mean Opinion Score} = \frac{\sum_i Opinion\ Score\ of\ Participant\ \ i}{Total\ Number\ of\ Participants}$$

Using this procedure, the mean opinion score curves using the 13 conducted experiments (1 to 2 samples per network impairment setting) are plotted in Figure A1 and A2 for the case when the non-congested participant and the congested participant is the game host respectively.

In the future, a proportional weight criterion will be explored in computing the mean opinion score where each participant's experience level determines how important the corresponding participant's opinion scores are.

Table A1. Participant Experience Level

Value	Response
1	Plays 0-20 hours/week but not FPS/MMOG games
2	Plays 21-40 hours/week but not FPS/MMOG games OR Plays 0-20 hours/week of FPS/MMOG games
3	Plays 41-60 hours/week but not FPS/MMOG games OR Plays 21-40 hours/week of FPS/MMOG games
4	Plays 61-80 hours/week but not FPS/MMOG games OR Plays 41-60 hours/week of FPS/MMOG games
5	Plays 81+ hours/week but not FPS/MMOG games OR Plays 61+ hours/week of FPS/MMOG games

Table A2. Participant Gameplay Satisfaction

Value	Response
0	Very Dissatisfied
1	Dissatisfied
2	Neutral
3	Satisfied
4	Very Satisfied

Table A3. Participant Gameplay Frustration

Value	Response
0	Very Frustrated
1	Frustrated
2	Neutral
3	Somewhat Frustrated
4	Not Frustrated

Table A4. Perceived Impact of Lag on Gameplay

Value	Response
0	A lot
1	Quite a Bit
2	Somewhat
3	Very Little
4	No Impact

Table A5. Participant's Likeliness to Change Service Providers

Value	Response
0	Very likely
1	Somewhat Likely
2	Neutral
3	Unlikely
4	Very Unlikely

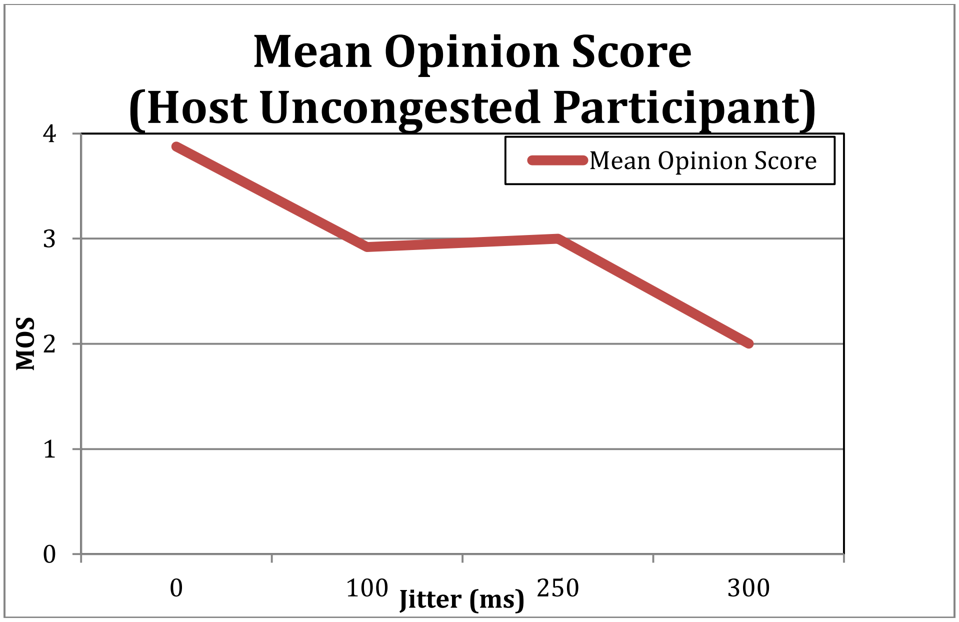


Figure A1. Mean Opinion Score when Uncongested Participant is Game Host

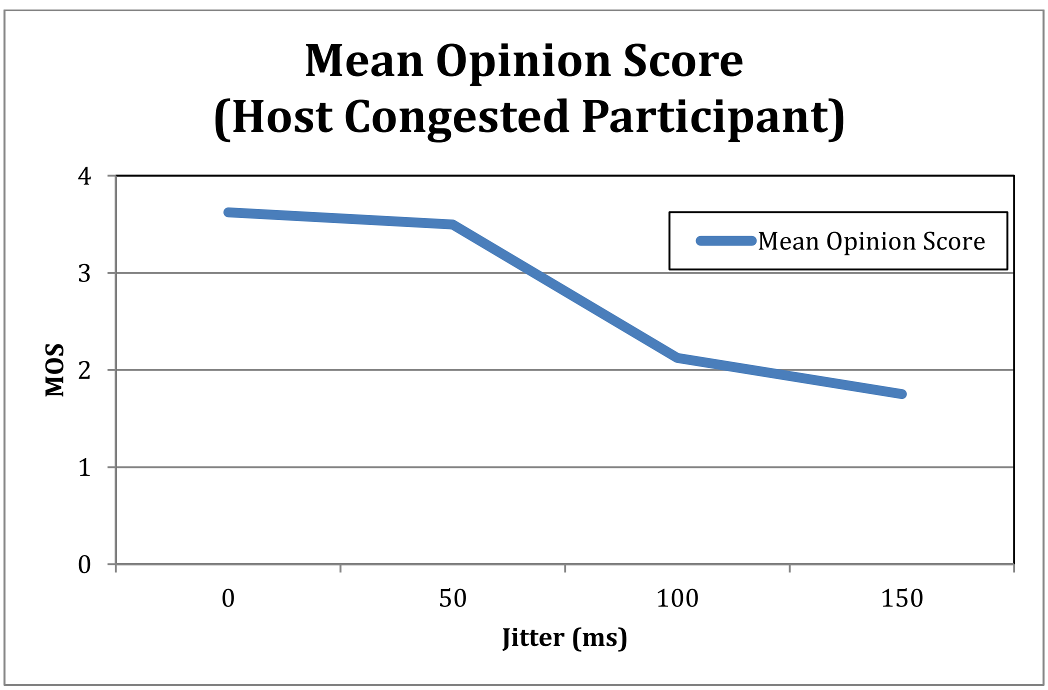


Figure A2. Mean Opinion Score when Congested Participant is Game Host