Limitations of using Real-World, Public Servers to Estimate Jitter Tolerance Of First Person Shooter Games

Grenville Armitage
Centre for Advanced Internet Architectures
Swinburne University of Technology
Melbourne, Australia
garmitage@swin.edu.au

Lawrence Stewart
Centre for Advanced Internet Architectures
Swinburne University of Technology
Melbourne, Australia
lastewart@swin.edu.au

ABSTRACT
This paper quantitively evaluates the relevance of network jitter on player satisfaction and performance in multiplayer online games, particularly in comparison to overall network latency (or 'ping') times. We extend previously published work that showed Quake III players preferred servers less than 150-180ms away. Our modified, public Quake III server logged 20+ latency samples per second per client, from which we derived the jitter (instantaneous latency fluctuations) for each connected player. We found that using real-world traffic resulted in a highly correlated relationship between jitter and latency (jitter being generally one fifth, or less, of the path's latency), making it difficult to derive any independent relationship between jitter and player satisfaction. However, our results do demonstrate that absolute jitter over typical Internet paths is far less significant than the absolute latency for interactive games such as Quake III. This suggests Internet service providers (ISPs) should focus primarily on bringing latency within reasonable bounds.

Categories and Subject Descriptors
C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design - *network communications, packet-switching networks.*

C.2.2 [**Computer-Communication Networks**]: Network Protocols – *applications.*

C.2.3 [**Computer-Communication Networks**]: Network Operations - *network monitoring.*

C.2.4 [**Computer-Communication Networks**]: Distributed Systems - *client/server, distributed applications*.

General Terms
Measurement, Performance, Experimentation, Human Factors.

Keywords
Jitter, network games, Quake III Arena, first person shooter, internet service provider

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGCHI ACE2004, June 3-5, 2004, Singapore.
Copyright 2004 ACM 1-58113-882-2/04/0006…$5.00.

1. INTRODUCTION
Networked, multiplayer games pose a challenge for Internet service providers and game server hosting organizations. Game play and game design techniques are being studied in a range of fora, whereas the interaction between game experience and network conditions is less well understood. Hosting companies need to understand the relationship between network conditions (such as latency, packet loss and jitter) and their reachable target markets to correctly focus marketing and customer support. ISPs need to understand how (or if) they should refine their network provisioning and engineering efforts to better cater to this new market segment. Our research aims to provide insights useful to both ISPs and organizations who host game servers.

Interactive online games have traffic profiles [2][5][12] and network demands quite unlike traditional web surfing, email and non-interactive peer-to-peer streaming applications. High latency negatively impacts on the game's interactivity. Packet loss either directly impacts on the game experience, or (if the application uses retransmission to cover-up packet loss) causes temporary increases in latency. Game traffic also looks distinctive – mostly regular streams of modest sized (60 to 250+ byte) packets, rather than the bursts of small or large packets typical of web page downloads, email transfer and other file transfer applications.

This paper specifically focuses on the likely relevance of network jitter on player experience with a fast paced, multiplayer “first person shooter” (FPS) game. Previously published research by one of the authors (Armitage) looked at the latency sensitivity of online Quake III players [1] without specifically looking at the possible impact of jitter (instantaneous, and usually random, fluctuations in network latency) on the player experience. We have now continued that research to report on jitter sensitivity of Quake III players, using a public server running from 14th November 2001 to 26th February 2002.

We found that using real-world, public game servers to gather statistically valid data is fraught with difficulty. The tight correlation between experimentally observed jitter and latency, makes it difficult to draw strong conclusions about the sensitivity of players to jitter alone. However, we conclude that jitter is far less of an issue than absolute latency - by the time a player observes notably high jitter their absolute latency is so bad the game would be unplayable anyway.

The rest of this paper describes the background to this research, the public server's configuration and location, our approach to extracting jitter and latency estimates from server logs, and our interpretation of the results.

2. BACKGROUND

Our goal is to extend previously published research [1] which found that online Quake III players tend to prefer online servers less than 150-180ms away. This research left open questions concerning the effects of jitter on a game player's experience independent of absolute latency.

A preliminary study of Half-Life in 2001 concluded that network latency was less important than vagaries in the latencies induced by application/host behaviors and the relative delays between players on the same server [8]. A more recent 2002 study with Warcraft3 found players could tolerate substantial fractions of a second in network latency [13]. Related work has tended to focus on modeling of network and user behaviors [5][14][9][17] rather than latency sensitivity per se.

Other research on the effects of latency and jitter on realtime applications have tended to focus on quality of service (QOS) issues and services like streaming video [3] and voice over IP [16]. There has been research using human perception trials to try and ascertain the levels at which latency and jitter become noticeable and debilitating in realtime networked applications [11]. There were no concrete figures specified for jitter, but it was clear that high latency with any jitter made visual coordination tasks almost impossible to complete.

Jitter comes from various sources – e.g. congestion-related queuing in core routers and serialization delays over rate-capped access links [10]). The impact is highly dependent on both the type of game (genre and playing style) and the architecture of the networking code. Recent games try to even out the latency differences between clients using internal lag-compensation algorithms., which in turn affect how a game is perceived by the player and their tolerance to changes in the network conditions.. Nevertheless, we believe the insights in this paper are generally applicable.

3. METHODOLOGY

In late 2001 we re-instrumented the Quake III server previously used to generate latency sensitivity insights [1] in order to capture many more latency ('ping') samples per second. The server's current network latency estimate to a given client was now sampled each time a client 'frame' (packet) arrived at the server. During active play Quake III clients transmit between 20 and 100 frames per second. In order to minimise local disk load, the server would accumulate these samples internally and dump a histogram to the server's on-disk logfile every 2000 client frames. (The server modification is available as gmod1.1 [7].)

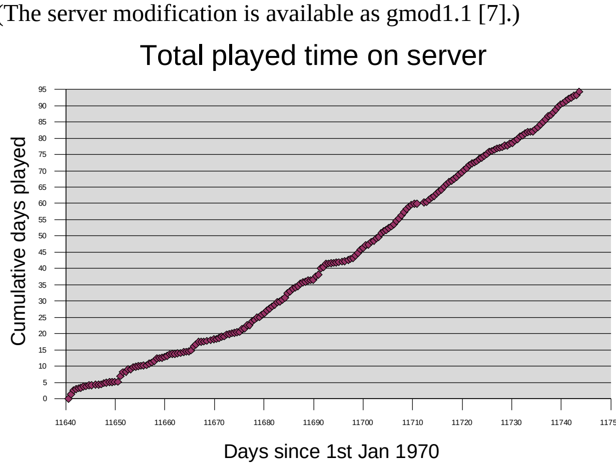

Figure 1. Total played time on server

The server ran from 14th November 2001 to 26th February 2002 in Palo Alto, Calforinia with a T1 link direct to PAIX (Palo Alto Internet eXchange). It was an Intel CA810e motherboard, 600MHz Celeron, 128MB RAM, running FreeBSD 4.4 [6]. We ran version 1.30 dedicated Quake III server, 6 maps, fixed cycle sequence, 15 minutes per map, and up to 6 remote players (and 2 bots). The server attracted 1936 unique [1] players who accumulated 94 days of played time over this period (Figure 1 shows the server being frequented regularly over this period).

3.1 Terminology

The following terms will be defined for the context of this paper:

Ping histogram: Logged after 2000 client frames have been seen or end of map or client disconnect. A frequency distribution of the ping values seen for the particular client since the previous ping histogram was logged.

Jitter: One standard deviation of a client's ping histogram. (For example, a jitter of 4ms would imply that 67% of all latency samples for a given histogram fell between ±2ms from the mean.)

Latency: The mean calculated from a given ping histogram.

Map: A virtual arena/area on which players do battle. Players on the server at the completion of a map were automatically placed on the next map in the sequence to continue playing.

Game: A period of time a client remained on the server, spanning one or more maps. If there are 180 seconds or more between the Nth and (N+1)th Ping histograms, the Nth histogram signifies the end of one game and the (N+1)th histogram represents the beginning of a new game.

3.2 Server calibration

We calibrated the Quake III server's internal ping estimation algorithms on an isolated network. The server ran a single map cycle, and all traffic to and from the server was piped through FreeBSD's dummynet [4] kernel module (Dummynet is used to add artificial latency to traffic and has been shown to be quite precise down to 1ms granularity with a suitably modified kernel [15].) A number of clients on the same subnet as the server played a single map cycle with dummynet configured for round trip latencies of 50ms, 100ms, 200ms and 300ms. A separate machine on the same subnet concurrently tested the network-level round trip time using standard ICMP Echo/Response

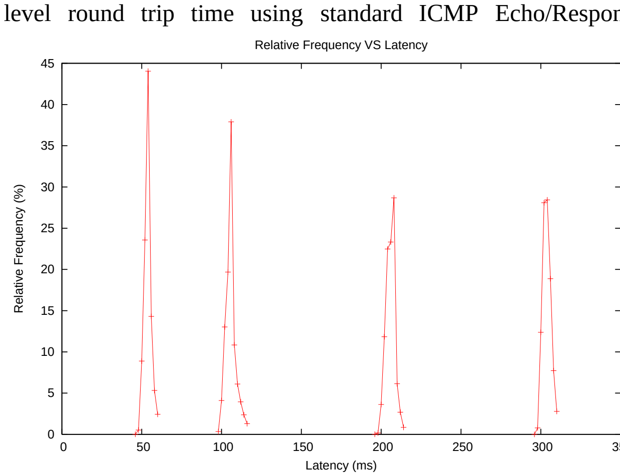

Figure 2. Typical Relative Frequency VS Latency as perceived by the Quake 3 server for clients

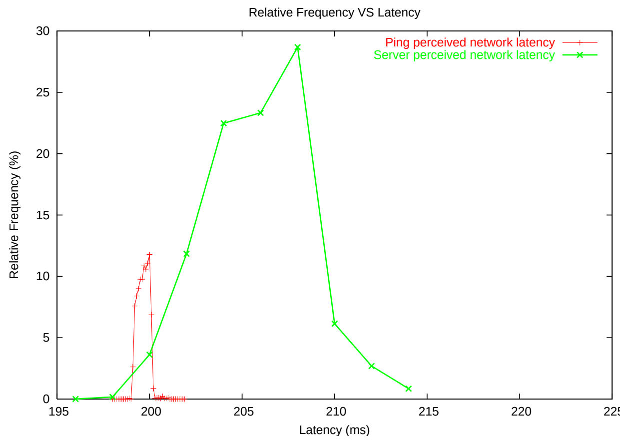

Figure 3. Typical Relative Frequency VS Latency as perceived by ping and the Quake 3 server for clients

('ping') packets every 500ms (consistently estimating the network layer latency within 1ms of that configured through dummynet).

We plotted the server's Ping histograms to characterise the spread of jitter and mean latency estimates per client. Figure 2 shows the spread of mean latency for the clients as reported by the Quake III server. Figure 3 shows an enlarged plot of the 200ms latency band of Figure 2, with the network ping reported round trip times shown in red for comparison. The server's latency estimation spreads from the actual configured latency to about 10ms-15ms above this value, with the average error around +5ms.

Figure 4 shows the spread of server-estimated jitter for each dummynet latency setting, with the average jitter marked by large crosses .

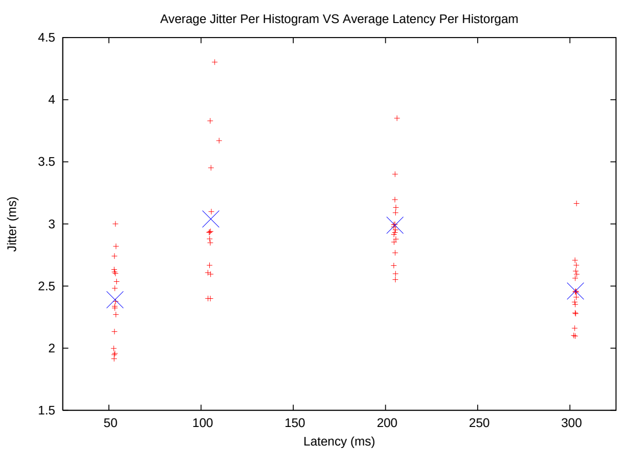

Figure 4. Typical Jitter VS Latency per histogram plot as perceived by the Quake 3 server for clients

At worst the server's application-level ping estimation algorithm overestimates latency by 15ms and add 4ms of jitter. On average, these values drop to 5ms of latency and 3ms of jitter. Thus we adjusted all our game-derived latency estimates down by 5ms, and jitter estimates are reduced by 3ms.

3.3 Selection of performance and satisfaction benchmarks

Performance in a first person shooter can be expressed in terms of frags (kills) per minute (frag rate) or deaths per minute. Analysis showed there to be no correlation between deaths per minute and either jitter or latency. However, there is a correlation for frags per minute, and so this will be a benchmark by which we judge player performance. In the absence of direct user feedback nidicating satisfaction with a game, we infer satisfaction based a client's time spent on the server.

3.4 Data considerations

Ping histograms whose latency and jitter fell in the lower 15% or upper 15% of all ping histogram samples were excluded from any averages presented in this paper. We ignored any player that did not accrue at least 15 minutes on the server, reducing the possibility of players skewing the frag rate results by jumping in, killing a few people, then leaving. In principle this could also have eliminated players with high latency and jitter who left early because of poor game experience. Fortunately in our case, the eliminated clients fell into the low latency, low jitter, high frag rate category. We also excluded (per map) any player who did not play at least 120 seconds on any given map, and who did not pick up at least one item per minute of gameplay. We considered more aggressive filtering (excluding players who picked up less than 5 items per minute, or less than 10 items per minute – see Table 1) but decided it was not necessary. The statistics for players who picked up between 1 and 9 items per minute showed basically the same distributions as for players who picked up at least 10 items per minute. Thus to maximise the number of data points available, the rest of our analysis is performed on players who picked up at least 1 item per minute.

	1 ipm	5 ipm	10 ipm
Unique players	1837	1808	1393
Games played	4931	4686	3999
Maps Played	11138	10550	7138
Frags made	220022	218837	170372

Table 1. Filtered statistics for 1, 5 and 10 items per minute

4. ANALYSIS

In-house tools where developed to post-analyse our Quake III server logs and extract player usage and latency/jitter statistics presented in the rest of this section (gmodstat [7] and gmodstatjitter [7]).

4.1 Jitter VS Latency

Jitter and latency are shown to be highly correlated in Figure 5. High latency can arise from large numbers of short hops (each of which introduce some queuing delay and associated jitter) and

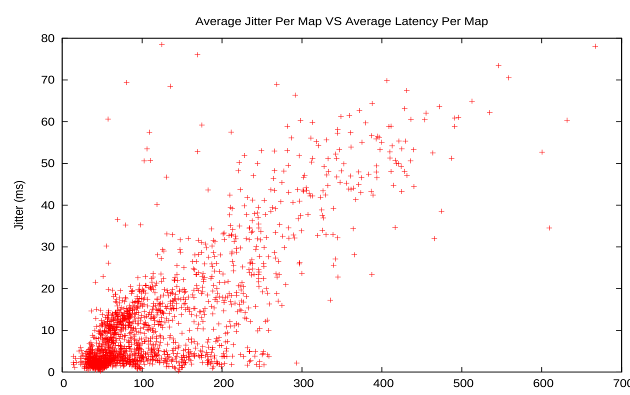

Figure 5. Average jitter VS Average latency per map

small number of large hops (where speed of light propagation is the major source of delay and there is very little jitter).

Figure 5 shows two distinct regions - one of low jitter points over a large spread of latencies (short numbers of long hops), the other of points that have jitter roughly linearly increasing with latency (multi-hop paths). The upper bound on latency is approximately 450ms with 60ms being the approximate upper bound on jitter.

Clearly it is difficult to rule out the effects of one variable on another, making analysis with respect to either latency or jitter alone extremely difficult.

4.2 Relative Jitter

Figure 6 shows a plot of normalised jitter (jitter divided by latency) vs latency. Normalised jitter tends to be roughly one fifth or less of any given latency. Above 260ms latency there are almost no data points whose normalized jitter is below 0.1 (suggesting that, from the perspective of our server in Palo Alto, paths over 260ms are predominantly high hop-count paths).

These results raise an interesting question: does the real world ever introduce enough jitter to be a problem? At latencies in the 150-180ms region (the upper tolerable bound identified in previous work [1]) jitter is consistently below 30ms. To experience jitter around 50ms you'd need latencies over 250ms, where we'd argue the playing experience is already pretty awful.

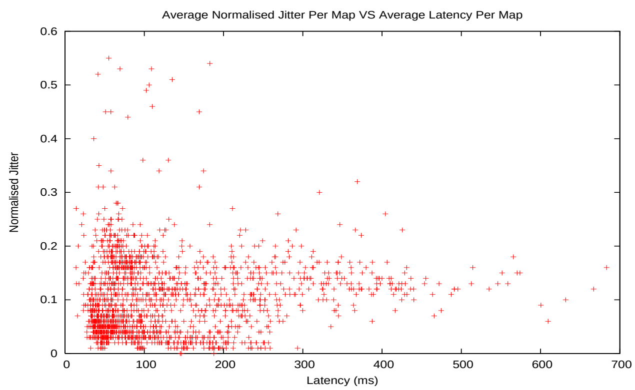

Figure 6. Average normalised jitter per map VS Average latency per map

4.3 It is all about the frags

We looked for correlation between frag rate and jitter. However, given the existing correlation between latency and jitter, it is difficult to eliminate the effect of latency on apparent player skill.

Figure 7 shows the average frag rate versus average jitter per map for each player. It is clear that jitter negatively impacts on frag rate, and that there are few data points above 70ms jitter. Figure 8 shows average frag rate across every map whose average jitter falls into 2ms jitter bands up to 70ms jitter. If we follow the linear trend line fitted to the data, we see that players with 0ms of jitter are fragging at a rate of 1 frag per minute more than people with 70ms of jitter. Over a 15 minute map, where a match is decided by a single frag, this creates a material difference in player success and satisfaction.

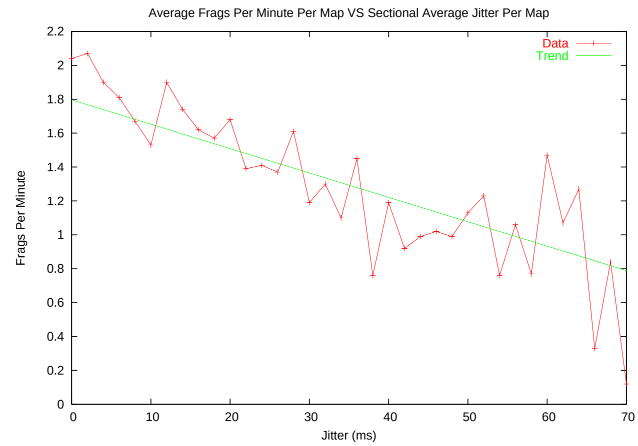

Figure 8. Average frags per minute VS Average jitter per map for 2ms jitter bands

We attempted to normalise frags per minute (FPM) against the number of players that were on the map - i.e. Normalised FPM = FPM/(number of players on map). (It should be easier to get a high frag count on maps with more players for the same skill level.) Higher normalised FPM thus implies greater skill. Figure 9 shows a plot of normalised FPM versus average jitter per map (in 2ms wide jitter bands), and lacks some of the exaggerated FPM values evident in Figure 7 at lower levels of jitter.

4.4 Isolating Latency and Jitter

In order to try and examine the effects of jitter independent of latency, we sliced the data into small latency bands and then analysed various game statistics in these bands. We also sliced the data into jitter bands and examined the effects of latency in these bands. The distribution of players who actually fell into

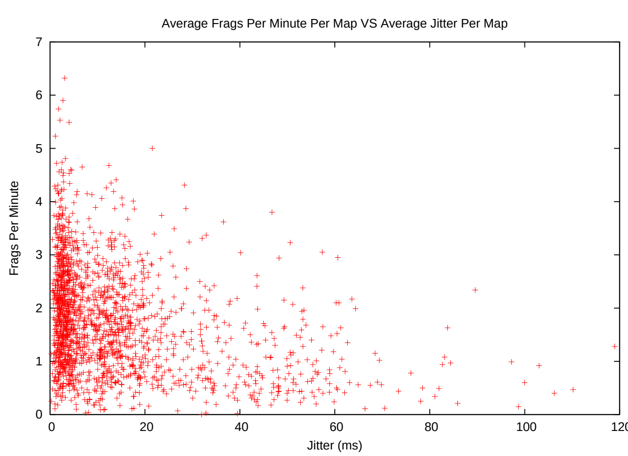

Figure 7. Average frags rate VS Average jitter per map

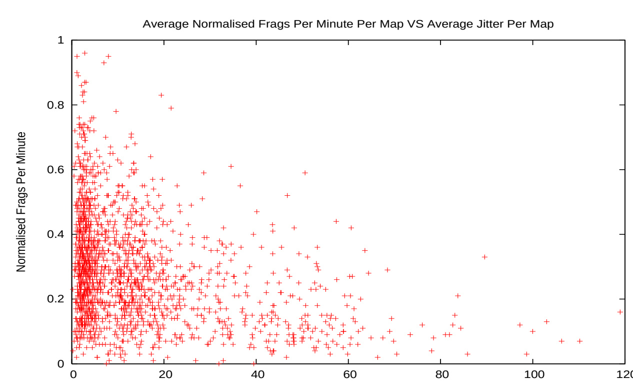

Figure 9. Average normalised frags per minute VS Average jitter per map for 2ms jitter bands

various latency bands is shown in Figure 10. The same information across jitter bands is shown in Figure 11.

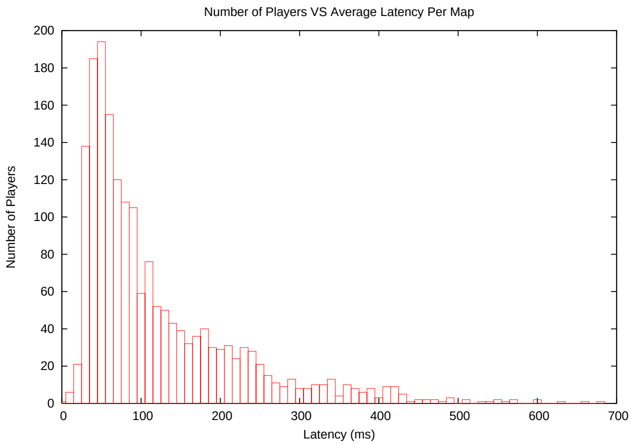

Figure 10. Number of players VS Average latency per map for 10ms latency bands

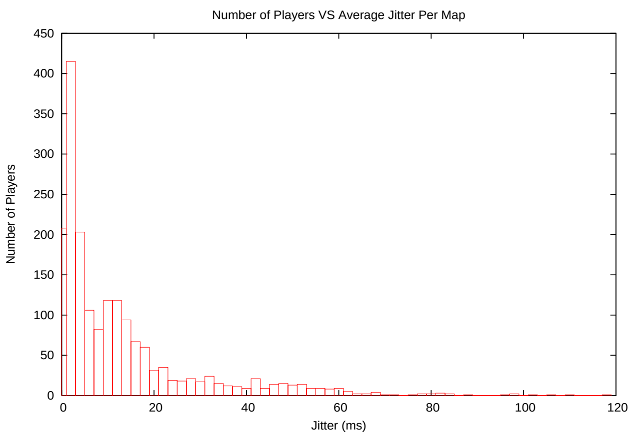

Figure 11. Number of players VS Average jitter per map for 2ms jitter bands

Based on the distribution of players, we chose the bands with the most players to give the most data points. Figure 12 shows the average FPM per map as a function of jitter for clients in the 50ms-60ms latency band (155 players). Figure 13 shows the average FPM per map versus latency for clients in the 2-4ms jitter band (which covers 415 players).

Figure 12 shows no strong evidence of jitter's negative impact on FPM. Given prior work [3][16] showing jitter does generally

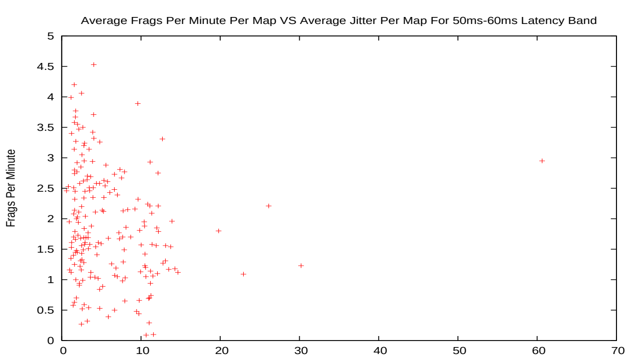

Figure 12. Average frags per minute VS Average jitter per map for 50ms-60ms latency band

negatively impact human reaction times, Figure 12 says there's no meaningful level of jitter around the most commonly observed latencies. (The range of observed jitters in the 50ms-60ms latency band is quite low. We lack a higher latency band with a larger jitter range and sufficient data points to analyse.)

With a large number of samples and broad range of observed latencies within the 2-4ms jitter band, Figure 13 shows modest evidence of FPM degradation with latency.

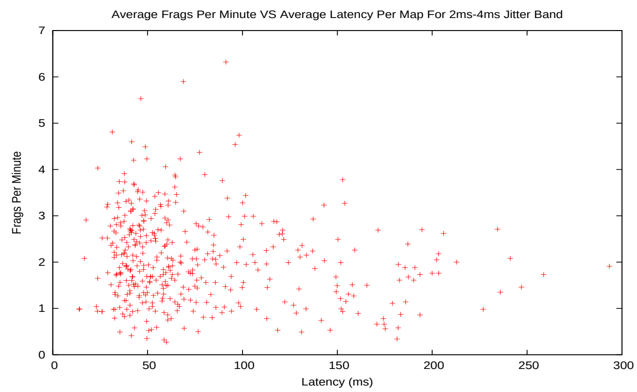

Figure 13. Average frags per minute VS Average latency per map for 2ms-4ms jitter band

4.5 Time spent on server

Figure 14 shows the average game length (normalised against the number of clients) versus jitter. The plot shows some evidence of decreasing average game length with increasing jitter, however, the experimentally observed correlation between jitter and latency makes it hard to discern the degree to which jitter itself affects a player's decision to keep playing or leave.

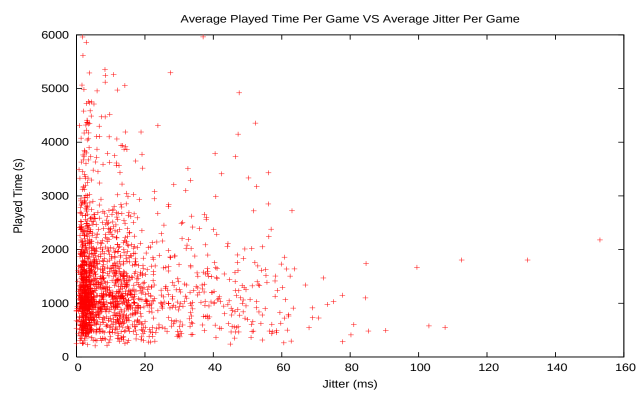

Figure 14. Average played time per game VS Average jitter per game

5. LIMITATIONS AND FURTHER WORK

This paper demonstrates that instrumenting a public, online Quake III server is a flawed method for ascertaining the independent contributions of latency and jitter to player satisfaction and success. It is flawed primarily because the Internet's current architecture creates a reasonably strong, observed correlation between latency and jitter. The server's logfiles represent the jitter/latency mix of players outside of our control. The absence of many high latency/low jitter or low latency/high jitter combinations playing on the server. makes it difficult to locate a broad spread of player experiences from which to draw strong conclusions.

Another weakness is our method of inferring player satisfaction. In the absence of players filling out feedback forms, we cannot be sure why a player leaves or decides to continue playing. We use frags per minute as an indicator of success, and attribute it to network conditions without being able to accurately account for differing player skill levels. Implicit in our analysis is the assumption that players of varying skills are distributed evenly across all latency and jitter ranges. We have no way to prove or verify this assumption. More experienced players will also tend to notice the adverse effects introduced by the network conditions more so than less experienced players. This has the potential to skew results depending on the player population that frequented the server.

A better approach in the future would be to run controlled user-perception trials, using specially constructed network testbeds having independently configurable latency and jitter characteristics. Such a trial would incorporate direct user/player feedback between maps and games, with which to screen and calibrate the objective measures such as frags per minute. With a controlled testbed we would be able to create jitter/latency combinations unobserved in the real Internet.

6. CONCLUSION

By setting up a specially instrumented, public Quake III game server we were able to log the latencies and jitters experienced by online clients between 14th November 2001 to 26 February 2002. Information covering 1837 unique players, 4931 games and 11138 maps was used to generate the results in this paper.

We observed that real-world clients coming in over the Internet saw rather strongly correlated jitter and latency – with the ratio of jitter to latency being almost entirely 0.2 (one fifth) or smaller. Two things fall out from this observation:

First, there are not enough data points distributed independently across the jitter and latency axes for us to derive the effect of jitter on player satisfaction independent of latency. Second, it appears that jitter over current Internet paths is actually insufficient to be of concern to online game players, even for highly interactive games such as Quake III.

Put another way, for any Internet path whose latency is tolerable in and of itself, the jitter will be too small to matter. For example, given that Quake III players prefer servers less than 150-180ms away [4] our results suggest they will be unlikely to see jitter above 30ms.

Clearly these results say more about the relationship between jitter and multi-hop paths than they do about Quake III player sensitivity to jitter. ISPs should continue to focus on keeping latencies bounded for their game-playing customers, as acceptable jitter will probably occur as a by product of current engineering practices.

Fellow (and future) researchers should not re-attempt this experiment on a public, uncontrolled game server. Future work on jitter sensitivity requires a controlled, isolated network environment where artificial jitter and latency can be varied independently and players can be asked to rate their playing satisfaction as trials progress.

7. ACKNOWLEDGEMENTS

Our thanks go to Dr Brian Reid for providing IP connectivity and rack space for the game server in Palo Alto during the months our server was running.

8. REFERENCES

[1] Armitage, G.J., "An Experimental Estimation of Latency Sensitivity In Multiplayer Quake 3", 11th IEEE International Conference on Networks (ICON 2003), Sydney, Australia, September 2003

[2] Borella, M. S., "Source models of network game traffic", Proceedings of networld+interop '99, Las Vegas, NV, May 1999

[3] Claypool, M. and Tanner, J., "The Effects of Jitter on the Perceptual Quality of Video", Department of Computer Science, Worcester Polytechnic Institute, http://www.cs.wpi.edu/~claypool/papers/jitter-pq/jitter-pq.ps (as of February 2004)

[4] dummynet, "Dummynet -- Traffic Shaper, Bandwidth Manager and Delay Emulator", http://www.FreeBSD.org/cgi/man.cgi?query=dummynet&sektion=4 (as of February 2004)

[5] Faerber, J., Network game traffic modelling , Proceedings of the 1st ACM workshop on Network and System Support for games, April 2002

[6] FreeBSD, http://www.freebsd.org (as of February 2004)

[7] GENIUS Project, "Data Gathering and Analysis Tools", http://caia.swin.edu.au/genius/genius-tools.html (as of February 2004)

[8] Henderson, T., Latency and user behaviour on a multiplayer games server. Proceedings of NGC 2001, London, UK, pp1-13, November 2001

[9] Henderson, T. and S. Bhatti, "Modelling user behaviour in networked games," Proceedings of ACM Multimedia 2001, Ottawa, Canada, pp212-220, October 2001

[10] Jehaes, T., Coppens, T., De Vleeschauwer, D., Van Doorselaer, B., Deckers, A., Naudts, W., Spruyt, K. and Smets, R., "Access Network Delay in Networked Games", Proceedings of NetGames 2003, Redwood City, California, pp55-62, May 2003

[11] Kenyon, R. V. and Park, K. S., ""Effects of Network Characteristics on Human Performance in a Collaborative Virtual Environment", Electronic Visualization Laboratory, University of Illinois Chicago, http://www.evl.uic.edu/cavern/hci/parkk.pdf

[12] Lang, T., Armitage, G. J., Branch, P. And Choo, H., "A Synthetic Traffic Model for Half-Life," Australian Telecommunications Networks & Applications Conference 2003, Melbourne (ATNAC 2003), Australia, December 2003

[13] Sheldon, N., E. Girard, S. Borg, M. Claypool and E. Agu, "The Effect of Latency on User Performance in Warcraft III," Technical Report WPICS- TR-03-07, Computer Science Department, Worcester Polytechnic Institute, March 2003

[14] Smed, J., Kaukoranta, T. and Hakkonen, H., "Aspects of Networking in Multiplayer Computer Games", Proceedings of International Conference on Application and Development of Computer Games in the 21st Century, Hong Kong, China, November 2001, http://www2.cs.utu.fi/staff/jouni.smed/papers/AspectsOfMCGs.pdf (as of February 2004)

[15] Vanhonacker, W. A., "Evaluation of the FreeBSD dummynet network performance simulation tool on a Pentium 4-based Ethernet Bridge", Technical Report CAIA-TR-031202A, Centre for Advanced Internet Architectures, Swinburne University of Technology, December 2003

[16] Venkateswaran, R., "An overview of VOIP and current research trends on QOS", www.tcs-america.com/Riverside/networkqos_iptel.doc (as of February 2004)

[17] Wu-chang Feng, Francis Chang, Wu-chi Feng, Jonathan Walpole., "Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server," SIGCOMM Internet Measurement Workshop, November 2002