

Examining User Performance in the presence of Latency and Jitter in Distributed Interactive Applications

Declan Delaney^φ, Paul Meenaghan^φ, Tomás Ward* and Séamus Mc Loone*

^φ*Department of Computer Science,
NUI Maynooth
Co. Kildare, Ireland
E-mail: ^φ decland@cs.may.ie*

**Department of Electronic Engineering,
NUI Maynooth
Co. Kildare, Ireland
E-mail: * tomas.ward@may.ie*

Abstract – Latency and jitter inherently limit the maintenance of consistency in Distributed Interactive Applications such as computer games, distributed whiteboards and real-time, collaborative environments. Although there has been much research into methods for maintaining consistency, there is a distinct lack of research exploring the connection between latency, jitter and the end user experience in Distributed Interactive Applications. We have developed an application that allows us to conduct trials under controlled latency and jitter conditions. This provides data, which can be analysed to characterise how people adapt to various degrees of latency and jitter. We present results that highlight how an increase in latency and jitter affect the end-user experience, thus confirming the need for techniques to combat latency and jitter in Distributed Interactive Applications. We also note that the effects of jitter are significantly greater than those of latency.

Keywords – Distributed Interactive Applications, Jitter, Latency.

I INTRODUCTION

Distributed Interactive Applications (DIAs) are a class of computer programs involving real-time users acting in a virtual environment to execute some collaborative, cooperative or interactive task. The distributed nature of such applications means that a consistent worldview across all participants is impossible to achieve. This results in each participant seeing slightly different events at different times. In such a case the end-user experience can be destroyed.

Although there are a number of issues that affect the consistency of a distributed application, the ultimate limitation is imposed by latency effects.

Latency has been defined in various ways [1-3]. In this paper we define it as the time taken from start of transmission of the first bit of a data packet at the application layer of one participating node to the receipt of the last bit of the same packet at the application layer of a second participating node. A number of delays combine to cause latency. There are queuing and processing delays at routers, bridges, gateways and at packet source and destination;

delays in transcoding the data (encryption / decryption and compression / decompression); propagation and transmission delays due to the speed of light and the speed of the communications link.

There has been a lot of research carried out into techniques and mechanisms to combat the effects of latency. Most of these techniques seek to reduce the number of packets that need to be transmitted between participants. By so doing, the latency associated with queuing, processing and transcoding is reduced. Some techniques include the following:

- Client-side prediction contract mechanisms: These include techniques such as dead reckoning [4, 5] and strategy-based modelling [6, 7].
- Relevance Filtering techniques: data is filtered based on criteria such as geographical proximity or rate of change [8]. Another form of relevance filtering is visibility culling [9].
- Network transmission protocol: Multicasting to groups of participants with the same data requirements. The Internet

offers very limited support to multicasting protocols [10].

- Packet bundling: Increase the packet size by sending multiple packets under the one header. This may also introduce latency if not used with care [8].
- Data Compression: Reduce the size of the information packet being transmitted, using for example, delta compression [8].
- Time Management: Synchronize events or wall clocks among participants. Events can be pre-empted or delayed to maintain consistency [11, 12].
- Priority Scheduling: Data essential to maintaining consistency is sent with the highest priority [13].
- Quality of Service (QoS) Techniques: network parameters such as bandwidth and latency are guaranteed by the service provider [14, 15].

Despite the research carried out on solving the latency problem, very little work has been carried out on the effects of latency variation or jitter on users of distributed applications. An interesting suite of experiments was performed by Vaghi et al. in which they examined the effects of gradually increasing latency on the interaction of two participants in a virtual ball game [16]. Their results indicated that the users modify their playing strategies as they attempt to cope with latency and they identified several types of adaptation strategies. The focus of this paper is on

examining the accuracy of user performance in the presence of latency. Furthermore, we extend our work to also include the effects of jitter. We intend to show that these effects can't be simply ignored.

In the following section we detail the development of the application that will be used to conduct experiments under controlled latency and jitter conditions. The experiments performed are outlined in section 3, while the results are presented and analysed in section 4. The final section indicates our conclusions and the future direction of the research.

II TEST PLATFORM

The test platform was designed to measure the ability of users to perform a single task in the presence of varying degrees of latency and jitter and at different speeds of operation. The application was developed in Java version 1.4.2 (java.sun.com), employing a Java class library for generating charts called JFreeChart version 0.9.16 (jfree.org/jfreechart/). The Eclipse Integrated Development Environment (IDE) version 2.1 was used to develop the application. The choice of java was made so that the application would be independent of the underlying operating system, thus allowing the application to be made available for multiple platforms over the Internet. The graphical user interface (GUI) was designed

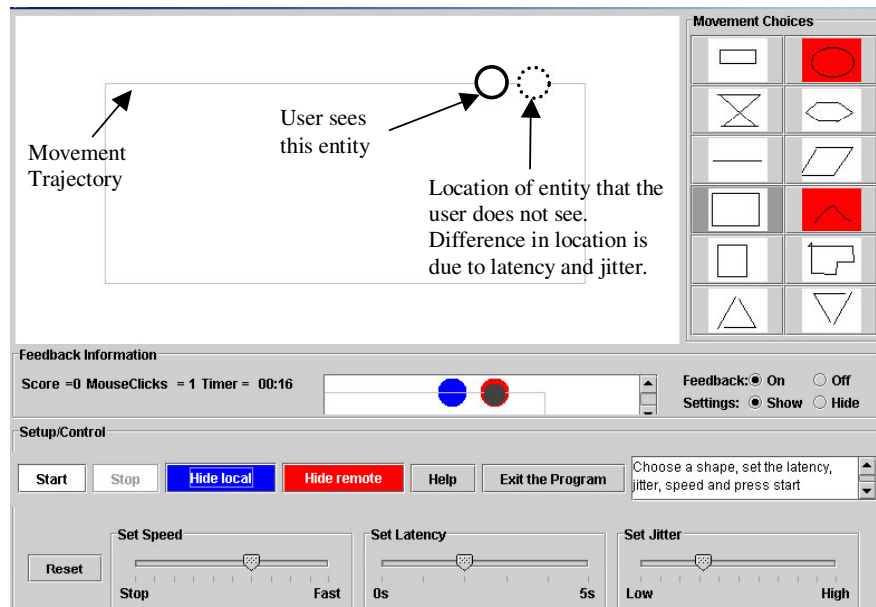


Figure 1: Graphical User Interface that is intuitive and allows the experimental parameters to be set.

with two objectives in mind: (1) to allow the experimenter set the experimental parameters easily and hide them from the user and (2) to provide an easily navigable layout and game-like environment with both audio and visual feedback. The GUI is shown in Figure 1.

The main features of the software are:

- Trajectory, latency and jitter values can be set using slider bars;
- Feedback to the user can be turned on/off;
- All experimental data is saved to a time-stamped directory in text format to facilitate export to other applications;
- A series of 15 plots are automatically generated when the user exits the application. A sample of one of these plots is shown in Figure 2.

The experimental platform simulates the local rendering of a remote entity (represented as a disk), which is moving along one of several possible trajectories. The user can only see the position of the disk computed locally, although the true position of the remote entity can also be shown if desired. The task performed by the participants in each experiment is to attempt to predict the actual location of the remote entity’s position under varying degrees of latency and jitter. All mouse clicks are recorded and when a correct ‘hit’ is made the application provides audio and visual feedback, as well as incrementing a hit counter. Although there is no limit to the number of mouse clicks that can be made, a hit can only be scored once for each simulation time unit.

The data produced by each experiment is recorded in ASCII format in a unique time stamped folder. In all 15 different variables are recorded such as time sample, hit count, jitter value, latency value, cumulative error and total mouse clicks.

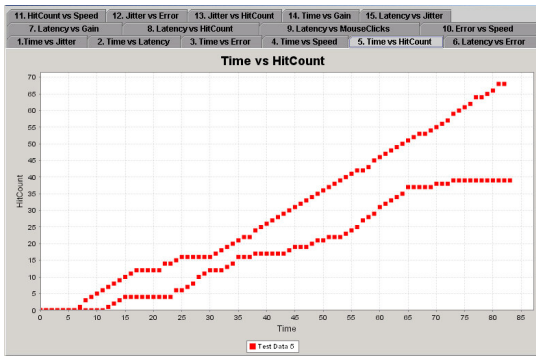


Figure 2: A series of graphs are automatically generated when the user exits the program – a sample graph is shown here.

The test platform was used to record data from a number of users. Details of these experiments are given in the following section.

III EXPERIMENTS

The test platform was employed to perform an experiment involving a total of eight volunteers. Each volunteer performed a single experiment consisting of a series of iterations during which the latency and jitter values were varied automatically by the application according to the parameters defined. The rectangular trajectory was employed. Users began at the top left-hand corner of the rectangle and moved in a clockwise direction. The latency and jitter values remained invariant for each iteration of the rectangle. A total of 12 individual iterations of the rectangle were performed with different setting for latency and jitter – see the list of iterations for the experiment in Table 1. Both audio and visual feedback were turned on but users were unaware of the latency and jitter parameters. The objectives of the iterations are described in the following paragraphs.

Iterations	Latency	Jitter
1	0	0
2	1	0
3	2	0
4	3	0
5	4	0
6	5 (max)	0
7	3	0
8	3	10
9	3	15
10	3	20
11	3	25
12	3	30 (max)

Table 1: A series of 12 iterations were performed. Jitter and latency were varied each time the user passed through the top left-hand corner of the rectangle.

Iteration 1: During this iteration we measure the participant’s performance when there is no jitter or latency. Both the local and remote positions are identical and the user only has to move the mouse cursor over the moving target and click. Each mouse click should result in a score with any errors being a result of user inaccuracy and not a result of latency or jitter. An error value of 11 or under results in a ‘hit’ or score. This value corresponds to the size of the object on the screen. This provides a benchmark or baseline measurement, against which other measurements can be accurately compared.

Iterations 2 to 6: Here jitter was kept at a constant value of zero and latency was increased in steps from

zero to its maximum value for the given application on every iteration of the rectangle. This provides a measure of the effects of latency in the absence of network jitter.

Iteration 7: This iteration is performed with a medium value of latency and zero jitter. This latency value will be maintained in the subsequent iterations. This iteration is identical to iteration 4 and marks the transition from latency-varying to jitter-varying iterations.

Iterations 8 to 12: In these iterations, latency is maintained at a medium value of 3 and user performance under progressively increasing jitter is measured. This provides data relating to the effects of jitter under conditions of constant latency.

An analysis of the results are presented in the following section.

IV ANALYSIS OF RESULTS

The presented results are preliminary and form the basis of understanding the behaviour of users under controlled conditions of latency and jitter. The results are shown for a typical user that is representative of all participants in the iteration.

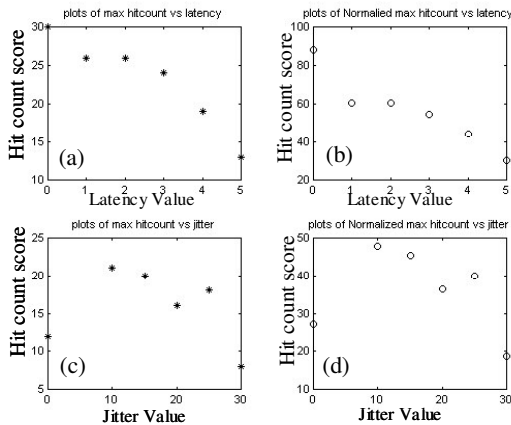


Figure 3(a)-(d): As latency is increased, the participant’s success at scoring a hit reduces. Introducing jitter also results in a significant reduction.

Figures 3(a)-(d) illustrate a representative performance of one of the participants in terms of hit count score as latency and jitter increase from 0 to their maximum values. Figures 3(b) and 3(d) are the normalized version of Figures 3(a) and 3(c) respectively. The normalization is performed with respect to the total number of mouse clicks made by the user. As latency increases, Figures 3(a) and 3(b) show a fall off in the score the user achieves, reducing from an absolute value of about 30 to under

15 when latency is at its maximum value. With latency maintained at a constant value, jitter tends to result in a poorer overall user performance with all scores lying between 10 and 20. In addition Figure 3(c) shows that the variation in scores is more random, compared to the gradual fall off in performance with increasing latency. It’s of interest to note that in Figure 3(a) when latency is 3 and jitter is zero, the hit count is 24. In Figure 3(c) when latency is reduced to 3 and jitter is still zero, the hit count is only 12. This was to be expected because the user would be expecting the trend of increasing latency to continue and the adjustment in latency was greater than between any of the previous iterations.

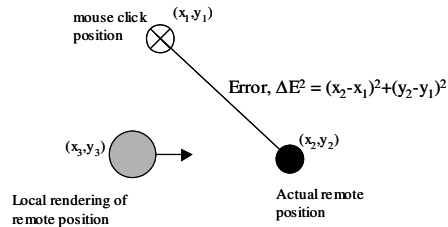


Figure 4: The instantaneous error is computed between the mouse click position and the actual position of the remote entity.

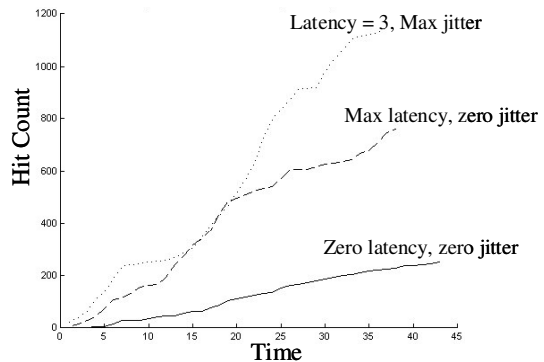


Figure 5: As time progresses the cumulative absolute error increases but does so more rapidly in the presence of jitter than that of latency.

The instantaneous error refers to the sum of the absolute distance from the position the mouse was clicked at to the actual position of the remote entity – see Figure 4. The cumulative error refers to the sum of the instantaneous error with time. A typical plot of cumulative error against time is shown in Figure 5 for three scenarios: (1) Jitter and latency both set to zero; (2) latency set to its maximum value and jitter set to zero and (3) latency at a medium value and jitter set to maximum. In scenario (1) the curve may be approximated as a straight line, indicating that the accumulation in error is a linear process. The other scenarios illustrate the rate of accumulation of error in the presence of high latency is more than doubled

compared to the case of zero latency and zero jitter. Finally we also note that there was a further increase in cumulative error due to the addition of jitter.

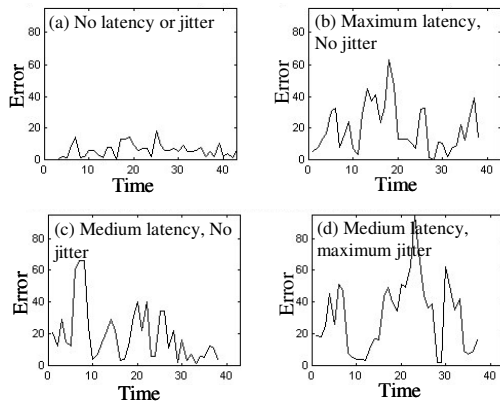


Figure 6(a)-(d): Plots of instantaneous error for different latency/jitter combinations

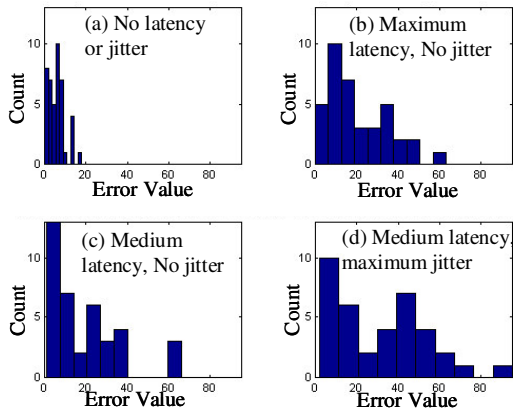


Figure 7: The distribution of errors increases with increasing latency and jitter.

Typical plots of the instantaneous errors as a function of time are shown in Figures 6(a)-(d). Figure 6(a) shows that with no latency or jitter the instantaneous error is low, usually below the experimental error level of 11. As latency and jitter are introduced the amplitude of error variation increases, with the worse case scenario being the case of medium latency and high jitter in Figure 6(d). An examination of the distribution of instantaneous errors clearly demonstrates the effects of jitter and latency on user performance - see Figure 7. The distributions become wider as latency and jitter are introduced. The distribution for zero latency and zero jitter has a maximum spread of about 20. In the case of medium latency and maximum jitter this increases to a spread of almost 100.

The average instantaneous error for each iteration was calculated together with the variance. These

values were plotted and are shown in Figure 8. The average error increases steadily and becomes erratic when jitter is introduced. The variance is also much less predictable when jitter is introduced. It is worth observing that users find it easier to adapt to constantly increasing latency but struggled to cope with the effects of jitter as indicated in Figure 8.

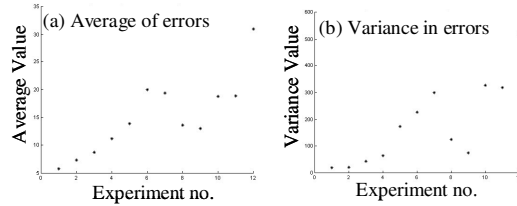


Figure 8: The variation in average error value and variance for each experiment

It is also interesting to note the importance of feedback to the user during the experiment. At low values of latency and jitter, users adapted to their presence by relying on the audio feedback. However, at high values of jitter they relied more on the visual feedback mechanism illustrated in Figure 9 because they could hear that they weren't making any hits and they needed some additional information to help them compensate. Visual feedback is provided in a small indicator screen showing the position of the user's actual mouse click and a line from there to the correct scoring position. It is obvious that feedback becomes more important at high latency and jitter values, but this is to be expected as the user's task becomes more difficult as latency and jitter are introduced. These findings confirm our intuition and are not altogether unexpected.

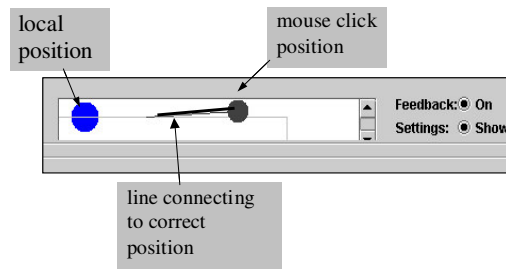


Figure 9: The Visual Feedback Mechanism.

V CONCLUSIONS and FUTURE WORK

A test platform and an experiment designed to allow us determine the effects of latency and jitter on user performance have been described. While the results presented in this paper are for one user only, it was found that all eight users showed similar behaviour. They illustrate that user performance, even in a relatively simple application, deteriorates rapidly in

the presence of latency and jitter and thus these factors cannot be simply ignored. The increase in research of latency masking techniques therefore needs to be matched by research into combating the effects of jitter.

The experiments conducted to date indicate a number of new avenues to explore. Several features will be added to the test platform such as recording of user mouse click coordinates, use of a scripting language to set up the experiments and increased size of the visual feedback screen. Our objective is to determine acceptable levels of jitter and latency values so that the user's psycho-perceptual experience remains acceptable.

REFERENCES

- [1] J. C. de Oliveira, S. Shirmohammadi, and N. D. Georganas, "Collaborative Virtual Environment Standards: A performance Evaluation," presented at Proceedings of the 3rd IEEE International Workshop on Distributed Simulation and RealTime Applications, October 23-24, College Park, Maryland, 1999.
- [2] S. K. Singhal and M. Zyda, *Networked Virtual Environments*. New York: ACM Press, 1999.
- [3] J. Smed, T. Kaukoranta, and H. Hakonen, "Aspects of Networking in Multiplayer Computer Games," presented at Proceedings of International conference on application and development of computer games in the 21st century, China, 2001.
- [4] IEEE, "IEEE Standard for Distributed Interactive Simulation - Application Protocols," IEEE 1995.
- [5] C. Durbach and J. M. Fourneau, "Performance Evaluation of dead reckoning mechanism," presented at Proceedings of the 2nd IEEE International Workshop on Distributed Interactive Simulation and RealTime Applications, May 1-4, Atlanta, Georgia, 1999.
- [6] D. Delaney, T. Ward, and S. Mc Loone, "On Reducing Entity State Update Packets in Distributed Interactive Simulations using a Hybrid Model," presented at Proceeding of the 21st IASTED International Multi-conference on Applied Informatics, February 10-13, Innsbruck, Austria, 2003.
- [7] D. Delaney, T. Ward, and S. Mc Loone, "Reducing Update Packets in Distributed Interactive Applications using a Hybrid Model," presented at 16th International Conference on Parallel and Distributed Computing Systems, August 13-15, Reno, USA, 2003.
- [8] M. A. Bassiouni, M. H. Chiu, M. Loper, M. Garnsey, and J. Williams, "Performance and reliability analysis of relevance filtering for scalable distributed interactive simulation," *ACM Transactions on modeling and computer simulation*, vol. 7, pp. 293-331, 1997.
- [9] C. Faisstnauer, D. Schmalstieg, and W. Purgathofer, "Priority Round-Robin Scheduling for very large Virtual Environments," presented at Proceedings of the IEEE Virtual Reality 2000 Conference, March 18-22, New Brunswick, New Jersey, 2000.
- [10] J. M. Pullen, "Reliable Multicast network Transport for Distributed Virtual Simulation," presented at Proceedings of the 3rd IEEE International Workshop on Distributed Simulation and RealTime Applications, October 23-24, College Park, Maryland, 1999.
- [11] B. G. Worthington and D. J. Roberts, "Encapsulating Network Latency Compensators in VRML," presented at Proceedings of VWSIM (Virtual Worlds and Simulation Conference), San Diego, USA, 2000.
- [12] D. J. Roberts and P. M. Sharkey, "Maximising Concurrency and Scalability in a Consistent , Causal, Distributed Virtual Reality System, whilst minimising the effect of Network Delays," presented at Proceedings of the 6th Workshop on Enabling Technologies Infrastructure for Collaborative Enterprises, Cambridge, MA, 1997.
- [13] C. Faisstnauer, D. Schmalstieg, and W. Purgathofer, "Scheduling for very large Virtual Environments and Networked Games using Visibility and priorities," presented at Proceedings of the 4th IEEE International Workshop on Distributed Simulation and RealTime Applications, August 24-26, San Francisco, California, 2000.
- [14] A. S. Tanenbaum, *Computer Networks*, 3 ed: Prentice Hall, 1996.
- [15] C. Chassot, A. Lozes, F. Garcia, M. Diaz, L. Dairaine, and L. Rojas, "QoS required by a DIS application in a new generation Internet," presented at IEEE Spring Simulation Interoperability Workshop, March 14-19, Orlando, Florida, 1999.
- [16] I. Vaghi, C. Greenhalgh, and S. Benford, "Coping with inconsistency due to network delays in collaborative virtual environments," presented at Proceedings of the ACM symposium on Virtual reality software and technology, London, England, 1999.