Does Reducing Packet Transmission Rates Help to Improve Consistency within Distributed Interactive Applications?

Damien Marshall, Séamus McLoone, Tomás Ward and Declan Delaney

Department of Electronic Engineering,
National University of Ireland Maynooth,
Maynooth, Co. Kildare, IRELAND
E-mail: dmarshall, seamus.mcloone, tomas.ward@eeng.nuim.ie

ABSTRACT

Networked games are an important class of distributed systems. In order for such applications to be successful, it is important that a sufficient level of consistency is maintained. To achieve this, a high level of network traffic is often required. However, this can cause an increase in network latency due to overloaded network hardware, which, ironically, can have a negative impact on consistency. Entity state prediction techniques aim to combat this effect by reducing network traffic. Although much work has focused on developing predictive schemes, there has been little work to date on the analysis of their true impact on the consistency of the system overall. In this paper, we identify an important performance-related characteristic of packet reduction schemes. It is demonstrated that there exists an optimal packet transmission region. Increasing or decreasing network traffic above or below this level negatively impacts on consistency. Based on this characteristic, it is proposed that predictive schemes exploit this optimal point in order to maximise consistency by efficiently utilising the available resources.

Keywords – Distributed Interactive Applications, Networked Games, Entity Update Mechanisms, Dead Reckoning, Consistency

I INTRODUCTION

Network games, a part of a larger class of applications known as Distributed Interactive Applications (DIA), have become extremely popular and commercially successful in recent years. At their core, they involve multiple participants collaborating and competing within a virtual environment, even though those participants may be located at geographically separate locations. Popular examples of networked games include first person shooters, such as Counter Strike, and massively multiplayer online games, such as World of Warcraft.

In order for interaction between participants within the virtual environment to be fruitful, it is important that a sufficient level of consistency is maintained in real time. Consistency is the degree to which each participant experiences the same worldview [1-2].

Achieving a fully consistent world state generally requires a large amount of network traffic as changes to the world made locally need to be communicated to all participants. However, excess network traffic can overload the underlying network connecting participants, resulting in increased network latency, jitter and packet loss due to swamped network hardware [3]. This then negatively impacts on world consistency as changes to the virtual world take extra time to propagate between participants.

Other work has focused on solving this issue, examining means of reducing network traffic, thus improving network conditions, while still maintaining a sufficient level of consistency. One popular method of doing this involves predictive models. Under this approach, information regarding entity dynamics, such as acceleration and velocity, are transmitted between participants. Each participant then uses this data to model the future behaviour of other participants. An update is not transmitted until the model and actual behaviour differ by a predefined error threshold. Examples of this approach include dead reckoning and the Hybrid Strategy Model [4-5]. In both cases, spatial difference is traditionally used as the error threshold metric.

In such techniques, there is a tradeoff between the error threshold and the underlying consistency that can be achieved. Lowering the error threshold, for example, will increase consistency as the model will be more accurate due to increased updates. Ironically, this may cause a resultant decrease in consistency, as the resultant increase in network traffic negatively affects the performance of the network and application. To date, however, we have no firm understanding of this tradeoff.

In this work, we highlight a key performance characteristic related to this tradeoff. Using a testbed developed in an industry standard games engine known as Torque [6], the amount of network traffic transmitted between participants operating in a typical home environment is varied. It is demonstrated that there is an optimal region of packet transmission rate. Increasing transmission rates above this region introduces inconsistency due to latency caused by overloaded network hardware. Such a situation necessitates the use of packet reduction techniques such as dead reckoning. On the other hand, reducing the rates below this region means that available capacity is being underutilised, which causes inconsistency, due to less information being transmitted in the form of less packets.

Based on this result, we propose that packet reduction schemes such as dead reckoning should take this characteristic into account during their execution, so as to optimally use available resources.

The rest of the paper is structured as follows. In Section II, we present some brief background information on network latency and entity state predictive schemes. In Section III, the test network environment for the experiment is described, and results collected using this testbed are outlined and analysed in Section IV. We conclude the paper in Section V.

II COMBATING LATENCY

One of the greatest limiting factors in achieving consistency within DIAs is network latency [1-2]. Network latency is the time taken for data to travel between a transmitter and receiver on the same network [7]. The problem of dealing with latency in DIAs is confounded by the variation in latency over time, known as jitter [8].

The causes and effects of network latency and jitter within DIAs have been well researched [9-10]. There are 4 main factors that contribute to latency:

1. Propagation delay – time to transmit a packet over the wire. Subject to speed of light delays
2. Transmission delay – time taken to serialize the data onto the wire.
3. Nodal delay – time to decide on a route for a packet at each router
4. Queuing delay – delay caused by waiting at overloaded network hardware.

Techniques that tackle the issue of latency and jitter tend to primarily concentrate on the reduction of network traffic so as to ameliorate the effects of queuing delay. A popular method of achieving this is predictive schemes, such as dead reckoning and the Hybrid Strategy Model [2-3].

Much work has analysed the performance of various predictive techniques. Roberts et al. propose an alternative to the standard spatial error threshold used in dead reckoning, known as time-space error threshold [11-12]. It is demonstrated how this technique improves consistency in comparison to traditional dead reckoning, at the expense of increased network traffic. The pre-reckoning algorithm, proposed by Duncan et al., improves on standard dead reckoning by providing a technique that detects likelihood of breaches of the error threshold, resulting in improved consistency [13]. Lee et al. propose a scheme under which the spatial error threshold value is increased based on distance between entities, resulting in less network traffic being transmitted between participants that are not close to one another within the virtual world [14]. The Hybrid Strategy Model (HSM) improves the dead reckoning predictive model by taking long-term user behaviour of an entity within a set environment into account [15]. This long-term model is known as a strategy model, and can include information detailing how an entity navigates a winding corridor, for example. This approach was shown to require fewer packets than dead reckoning, whilst maintaining an equal or greater level of consistency.

All the approaches outlined above allow a certain level of inconsistency for a resultant decrease in network traffic. However, none of these approaches take the capabilities of the underlying network into account in their operation, meaning that such approaches may be performing sub-optimally. This issue arises, as there is no firm understanding of how the relative reduction or increase in network traffic transmitted will actually impact on the overall consistency of the virtual environment.

To analyse this issue, a number of network trials were conducted using an industry standard engine known as Torque. The network setup for these trials is described in the next section.

III EXPERIMENTATION

Each experiment was carried out over a LAN using two desktop computers. Both test computers had equal specifications. Each computer also ran a copy of Torque, which was modified in two key ways to facilitate the experimentation. Firstly, the default client/server architecture was modified to operate in a peer-to-peer fashion, in order to emulate the architecture of applications that use packet reduction techniques such as dead reckoning. Secondly, the engine was extended to include full logging of both local and remote client information.

Both clients were connected via a network bridge running NetDisturb software, as shown in Figure 1. Net Disturb allows for the emulation of Wide Area Network limitations, such as latency and queuing delays, within a Local Area Network environment [16].

Using Net Disturb, we emulated the capacity of a home ADSL connection with an upstream bandwidth of 256kb/s. However, given that only 2 participants were involved in each experiment, this would mean that each participant had a full 256kb/s bandwidth, which is unrealistic. To compensate for this, we scaled the bandwidth down to 14kb/s, so as to emulate the performance of hosting 18 distinct clients. No limit was set on the buffer size, meaning that no packets could be lost due to buffer overflow.

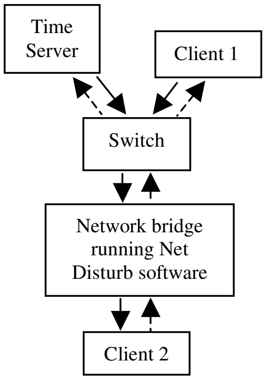


Figure 1. Overview of the network setup

Upon entering the virtual environment, each participant first contacted the other and initiated communication. Next, each participant synchronised his or her time with that of the "Time Server". This was done so as recorded information could be accurately compared offline following the experiments. The experiment then began. The goal in the environment was to navigate a racetrack, and reach the end position before the other participant. A plan view of the test scenario is shown in Figure 2. The

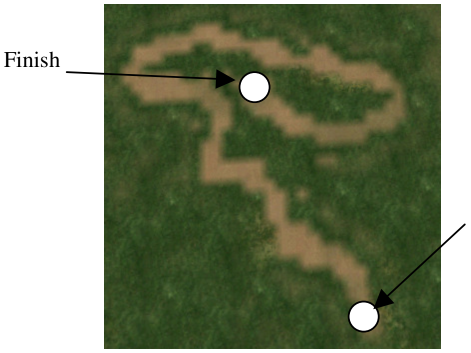


Figure 2 Plan view of test environment

packet transmission rate was varied per experiment, beginning at 2 packets per second (PPS), and incremented by 2 up to 24 PPS. All data collected during each experiment was then compared offline using Matlab. The spatial inconsistency, which is the Euclidean difference between the local and remote position of a client, was calculated at each time step. As it is impossible to guarantee that data will be recorded at the exact time on both machines, each local data point recorded was compared with that which has the closest time stamp. The maximum difference in compared timestamps was 2ms. The average spatial inconsistency was then calculated for the entire experiment.

IV RESULTS AND DISCUSSION

Analysing the data collected during the experiments described in Section III, the tradeoff between packet transmission rate and consistency is evident, as shown in Figure 3. Here, the average spatial inconsistency in Torque Game Units (TGU) is plotted against increasing packet transmission rates for a single set of participant trajectories.

As can be clearly seen from Figure 3, there is a region between approximately 18-22 PPS, within which average spatial inconsistency is at its lowest level. This is the optimal region of packet transmission rates, as increasing or reducing network traffic above or below this level only serves to negatively impact on consistency.

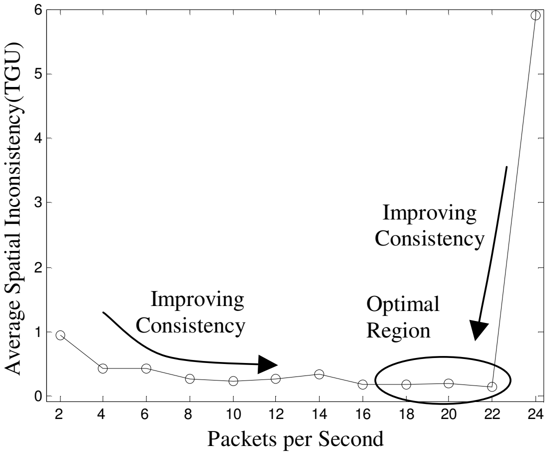


Figure 3. Spatial inconsistency measured as the packets per second increases

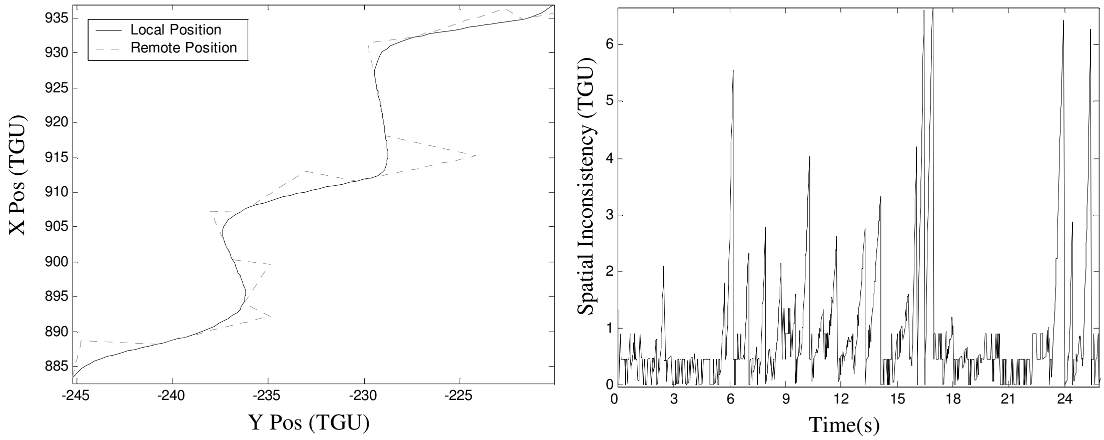
To the left of this region, inconsistency is being unnecessarily introduced due to modelling error. As the bandwidth being consumed within this region is well below that of the available 14kb/s, no extra network latency is caused by excess packet transmission. In such a case, it is worthwhile transmitting extra packets, so as to improve consistency. The typical behaviour exhibited within this region is shown in Figure 4. Here, a section of a single user local and remote path is shown when a transmission rate of 2PPS is used, along with the spatial inconsistency arising from a different section of the course. From Figure 4(a), it can be seen how the remote behaviour deviates from that of the local behaviour until an update is received, at which point, the two are reconciled. This is further evident from Figure 4(b). Note how the spatial inconsistency increases between updates, then returns to zero when an update is received.

Currently, predictive packet reduction schemes do not take this result into account. Their use, therefore, may result in an inefficient use of available resources. Based on this, we propose that such schemes incorporate a dynamic threshold, which can be scaled based on application and network parameters, so that packet transmission rates match that of the optimal region. The location of this optimal region will vary based on the nature of the application.

In the optimal region, the average bandwidth being consumed is approximately 13.5kb/s, which is just below that of the available bandwidth. Any increase above this level will overload the available bandwidth, leading to increased network latency. The impact of this increase on consistency is obvious from Figure 3, where it can be seen that inconsistency increases dramatically above 22 PPS. Consistency is improved in this case by reducing network traffic. Such a scenario necessitates the use of packet reduction schemes, such as dead reckoning. However, as demonstrated by the previous result, network traffic should not be reduced below the optimal region.

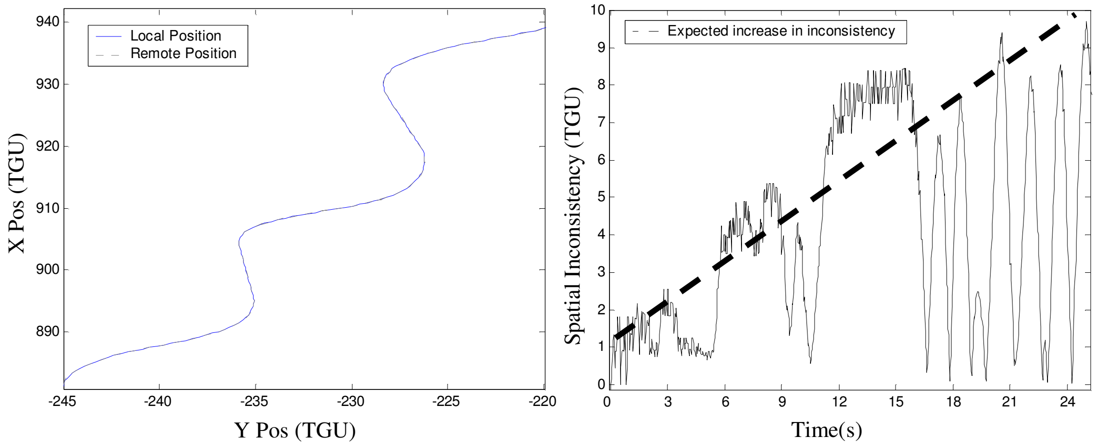
We further highlight the impact of network traffic at this end of the spectrum of transmission rates by analysing a single user trajectory, shown Figure 5(a) and (b). Figure 5(a) shows the local and remote trajectory for a packet transmission rate of 24PPS whilst figure 5(b) shows the spatial inconsistency arising from a different section of the course. Comparing Figure 5(a) and Figure 4(a), the effect of the extra packets is apparent. The remote behaviour is now indistinguishable from that of the local behaviour. However, examining the spatial inconsistency in Figure 5(b), the true impact of the higher transmission rates and network latency can be seen. In this case, the average bandwidth usage is 16kb/s, 2kb/s greater than the available bandwidth. However, as the NetDisturb buffer does not overrun, latency within the overloaded network continually increases, as the buffer grows larger. This has the knock on effect of continually increasing spatial inconsistency. Although this trend is evident from Figure 5(b), as shown by the broken straight line, there are occasions where the spatial inconsistency value decreases. The reasons behind this will now be discussed.

On the straight sections of the course, as shown in Figure 6, the remote position remains at a distance behind the local position due to network latency. The Euclidean distance measure used to calculate inconsistency works well in this case, and inconsistency has a value of *d*. As the



(a) (b)

Figure 4(a) An example trajectory showing the local and remote positions when a transmission rate of 2PPS is used (b) Spatial inconsistency for a section of the course



(a) (b)

Figure 5(a) An example trajectory showing the local and remote positions when a transmission rate of 24PPS is used (b) Spatial inconsistency for a section of the course

entity navigates a corner the distance around the curve between the two positions is still given by *d*. However, as we are using Euclidean distance, the inconsistency measured at this point is actually *d'*, which is smaller than *d*. Upon reaching the straight section of the track again, the inconsistency is once again correctly measured as *d*. This demonstrates a shortcoming in the spatial measure of inconsistency, a solution to which is outside the scope of this work.

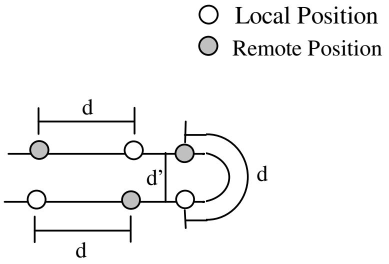


Figure 6. As the player navigates the course, the measured inconsistency can vary due to the nature of the inconsistency metric

V CONCLUDING REMARKS

In this work, the tradeoffs between network packet transmission rates and consistency were investigated, in order to analyse the true impact of packet reduction schemes on the performance of Distributed Interactive Applications. Using the Torque games engine, converted to operate in a peer-to-peer fashion, a number of network trials were conducted.

Results collected from these trials highlight an important performance related characteristic of packet reduction. There exists an optimal transmission rate region, within which average inconsistency is at it's lowest. Increasing the packet transmission rate above this point means that inconsistency is introduced due to network latency. Such a scenario requires the use of predictive schemes, such as dead reckoning. Reducing the rate below this level however, introduces inconsistency unnecessarily, without providing any subsequent improvement in network latency. Using these results, a shortcoming of the traditional spatial inconsistency measure was also demonstrated.

Based on this work, we propose that predictive schemes take this optimum packet transmission region into account, so as they can operate in an efficient manner. Such a scheme could use a threshold value, as with other predictive schemes. However, the value of this threshold could be scaled based on application attributes, in order to maintain a packet transmission rate that lies within the optimal region for that application. This is the focus of future work.

ACKNOWLEDGEMENTS

This work is supported by Science Foundation Ireland and Enterprise Ireland - grant no. IRCSET/SC/04/CS0289.

REFERENCES

[1] Delaney, D., T. Ward and S. McLoone. "On consistency and network latency in distributed interactive applications: A survey - Part I." *Presence: Teleoperators and Virtual Environments* 15(4),August 2006, pp 218-234.

[2] Delaney, D., T. Ward and S. McLoone. "On consistency and network latency in distributed interactive applications: A survey - Part II." *Presence: Teleoperators and Virtual Environments* 15(4),August 2006, pp 465-482.

[3] Roehle, B. (1997). "Channeling the data flood." IEEE Spectrum 34(3). p.p 32-38. 1997

[4] IEEE. IEEE Standard for Distributed Interactive Simulation - Application Protocols IEEE Std 1278.1-1995 IEEE, 1995

[5] Delaney, D., T. Ward and S. Mc Loone "Reducing Update Packets in Distributed Interactive Applications using a Hybrid Model". *16th International Conference on Parallel and Distributed Computing Systems, August 13-15*, Reno, USA, pp. 417-422, 2003.

[6] Lloyd, J. "The Torque Game Engine", Game Developer Magazine, 11 (8), pp. 8-9, September 2004.

[7] S. Singhal and M. Zyda, “Networked virtual environments: design and implementation.”, ACM Press SIGGRAPH Series, Addison-Wesley, Reading, Massachusetts, 1999.

[8] Blow, J.. "A look at Latency in Networked Games." Game Developer **5**(7): 28-40, 1998.

[9] Bolot, JC. “End-to-end packet delay and loss behavior in the internet” *SIGCOMM '93: Conference proceedings on Communications architectures, protocols and applications*, San Francisco, California. pp: 289 - 298

[10]Vaghi, I., C. Greenhalgh and S. Benford. Coping with inconsistency due to network delays in collaborative virtual environments. Symposium on Virtual Reality Software and Technology (VSRT '99), London, England, ACM, pp. 42-49.

[11] Roberts, D., D. Marshall, S. McLoone, D. Delaney, T. Ward and R. Aspin "Exploring the use of local inconsistency measures as thresholds for dead reckoning update packet generation". *Distributed Simulation and Real Time Applications*, Montreal, Canada, pp. 195 - 202, 2005.

[12] Zhou, S., W. Cai, F. B. S. Lee and S. J. Turner. "Time-space Consistency in Large Scale Distributed Virtual Environment." *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 14(1):pp. 31 - 47,2004

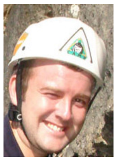
[13] Duncan, T. P. and D. Gracanin "Pre-reckoning algorithm for distributed virtual environments". *Winter Simulation Conference*, New Orleans, Louisiana, 2003, pp. 1086-1093.

[14] Lee, B.-S., W. Cai, S. J. Turner and L. Chen. "Adaptive Dead Reckoning algorithms for distributed Interactive Simulation." *International Journal of Simulation* 1(1-2):pp. 21-34,

[15] Delaney, D., T. Ward and S. Mc Loone "On Reducing Entity State Update Packets in Distributed Interactive Simulations using a Hybrid Model". *Proceeding of the 21st IASTED International Multi-conference on Applied Informatics, February 10-13*, Innsbruck, Austria,pp. 2003

[16] NetDisturb Website, ZTI http://www.zti-telecom.com/pages/main-netdisturb.htm. Accessed 21 Sept 2006.

AUTHOR BIOGRAPHIES



Damien Marshall graduated with a B.Sc degree (Computer Science and Software Engineering) in 2003. He then joined the Distributed Interactive Applications Group, and completed his M.Sc in January 2005. Shortly afterwards, he began his Ph.D. Currently, his work examines the various factors of consistency in distributed interactive applications.

Dr. Séamus McLoone is a member of the distributed interactive applications group (DIAG) at the Department of Electronic Engineering in the National University of Ireland Maynooth. His research interests varies from multiple model approaches to nonlinear system identification to reducing and masking the effects of latency in distributed interactive applications.

Dr. Tomas Ward is a member of the distributed interactive applications group at the NUI Maynooth. His main interests are in assistive technology and data management techniques for distributed shared applications. Spare time is spent on pushing out his personal best on Konami DrumMania V.

Declan Delaney received his B.E. and M.Eng.Sc. from University College Dublin in 1991 and 1998 respectively. His PhD was awarded in 2005 by the National University of Ireland, Maynooth. From 1992 to 2000 he worked as a software developer and subsequently as project manager. At present he works in the area of Knowledge management for an Italian multinational, ENI and is based in Milan,