REDUCING UPDATE PACKETS IN DISTRIBUTED INTERACTIVE APPLICATIONS USING A HYBRID APPROACH

Declan Delaney
Department of Computer Science, National University of Ireland, Maynooth, Co. Kildare, Ireland.
decland@cs.may.ie

Tomás Ward, Séamus McLoone
Department of Electronic Engineering, National University of Ireland, Maynooth, Co. Kildare, Ireland.
{tomas.ward, seamus.mcloone}@eeng.may.ie

ABSTRACT

Distributed Interactive Applications (DIAs) such as networked games or Distributed Interactive Simulations (DIS) depend on the underlying network. An important aspect is the number of data packets transmitted across the connected networks. To reduce the number of packets transmitted, DIAs employ client-side predictive contracts. The most commonly used client side predictive contract technique is pure dead reckoning. In this paper an alternative method in the form of a novel hybrid multiple-strategy technique is proposed. This technique chooses one of several models, either a deterministic dead reckoning model or one of numerous strategy-based models. This new approach is compared to the first order dead reckoning method using two game-type environments. It is shown that the proposed technique results in a significant reduction in the number of entity packets that need to be transmitted to maintain entity-state fidelity across the networked application.

KEY WORDS
Network Latency, Hybrid Models, Distributed Interactive Applications, Dead Reckoning.

1. INTRODUCTION

Distributed Interactive Applications (DIAs) such as networked games or Distributed Interactive Simulation (DIS) involve multiple participants communicating over a computer network [1][2]. By definition, a DIA must provide a realistic interactive experience to the participants. However, a number of technical problems combine to make delivery of such an experience difficult [3][4][5]. One major problem is network latency, which is the time it takes for information to propagate across the network to all participants. Another closely related issue is the problem of network bandwidth. Within a DIA these problems are referred to as the information updating issue. Several methods have been devised to reduce the quantity of data that needs to be transmitted between participants [6][7][8]. The DIS standard defines a client predictive contract mechanism called dead reckoning [9].

This paper proposes a hybrid multiple-strategy technique. This involves dynamically switching between one of several models, either a dead reckoning model or one of a number of strategy models. This technique results in a reduction in the number of packets that must be communicated to maintain remote fidelity of local entity movement compared to a purely dead reckoning contract.

The key contribution here is that a priori knowledge of user behavior is integrated into the proposed technique in order to reduce entity state update packets. In the absence of such reliable user-behavioral information, conventional dead reckoning protocols are employed. Such an approach is flexible, powerful and necessary for today's increasingly complex DIAs.

Two different game-type environments are used to illustrate our proposed technique, the results of which are compared to that of pure dead reckoning. Here, the hybrid multiple-strategy method dynamically selects the most suitable of four models based on dead reckoning and three strategy models. Results illustrate the validity of the new approach.

The next section outlines the information updating issue as it applies to Distributed Interactive Applications and describes existing solutions, including dead reckoning. Section three then describes our novel hybrid multiple-strategy approach. In section four we detail the construction of the multiple model for a purpose-built test environment. Results are presented in section five for dead reckoning, hybrid single-strategy and hybrid multiple-strategy techniques. The paper ends with conclusions and suggestions for future research.

2. THE INFORMATION UPDATING ISSUE

Network latency and bandwidth restrictions can combine to provide poor interactive experience in a distributed application. The Information Updating Issue refers to maintaining a consistent view of the shared environment among all participants of a Distributed Interactive Application [10].

The most common solution to the information updating issue involves a client-side prediction contract mechanism called dead reckoning [9]: all participating clients agree to maintain the same low order local models of the dynamics of all other participating entities. This is the contract. Each participant also maintains a model of its own entity dynamics, which it continuously compares to its actual dynamics. When these differ by a pre-defined threshold, update information is broadcast to all other participants. These then update their models for that entity. Convergence algorithms are necessary to allow a natural transition to occur between the modeled and actual motion when update data is received [11]. We do not deal with convergence algorithms in this paper.

Alternative methods have also been explored, and these include:

- Relevance Filtering techniques: These seek to reduce the information being transmitted over the network by filtering the data based on criteria such as geographical proximity or rate of change [8].
- Network transmission protocol: Multicasting and reliable multicasting allow hosts to subscribe and unsubscribe to any of possibly several multicast groups. Multicast groups might be created based on entity type or geographical location in the virtual environment [12].
- Packet bundling: This involves combining a number of data packets to create a larger data packet because network devices can only process a limited number of packets per unit time [8].
- Data Compression: These techniques allow the reduction in the size of the information packet being transmitted. One technique is to encode differences between successive data packets instead of transmitting the absolute state [8].
- Time Management: This involves pre-empting events and then locking up the system so that the event can occur. Alternatively, the execution of local user input is delayed and disguised as something else until the local user input can be relayed to all participants [13][14].
- Priority Scheduling: A transmission priority can be assigned to information based on criteria such as speed of movement or rate of error change [3]. This also includes Quality of Service protocols [5].
- Visibility Culling: The environment is divided into cells and multicasting updates are provided to all entities that are visible to each other in each cell [15].
- Quality of Service Techniques (QoS): QoS (Quality of Service) is the idea that transmission rates, error rates, and other network characteristics can be measured, improved, and, to some extent, guaranteed in advance. There have been attempts to define QoS measures suitable for DIS [16].

The above techniques are based on network management and network partitioning policies. We are going to look at a packet-reduction method based on client behavioral modeling, where we switch between a short-term dead reckoning model and a long-term statistical-based multiple strategy model. It should be pointed out at this juncture that our technique or indeed any other client-predictive contract can easily be complemented through the use of any of the techniques mentioned above.

3. THE HYBRID APPROACH

3.1 TERMINOLOGY

For the convenience of the reader we define the following terms. A *goal* is the aim or objective a person has in moving through an environment. For example, the goal might be to go from point A to point B. In achieving a goal a person can adopt a number of strategies, so that any one strategy is an expression of the goal. Strategies can be either steady state or transient. A *steady-state strategy* is a user-preferred method for achieving a set goal. A *transient strategy* is an exploratory method in an attempt to reach a steady-state strategy.

The idea underlying the hunt for strategies is to train a system to expect certain strategies based on past user behavior or based on expected user behavior. Each strategy comprises one or more trajectories – a set of trajectories can be associated with any strategy. A *trajectory* is a particular expression of a strategy. Multiplicity can refer to either strategies or goals. *Strategy Multiplicity Index* (SMI) refers to the number of goals a strategy leads to. *Goal Multiplicity Index* (GMI) refers to the number of steady-state strategies that lead to the goal. This terminology is illustrated in Figure 1. In this paper we present results for a GMI of 1.

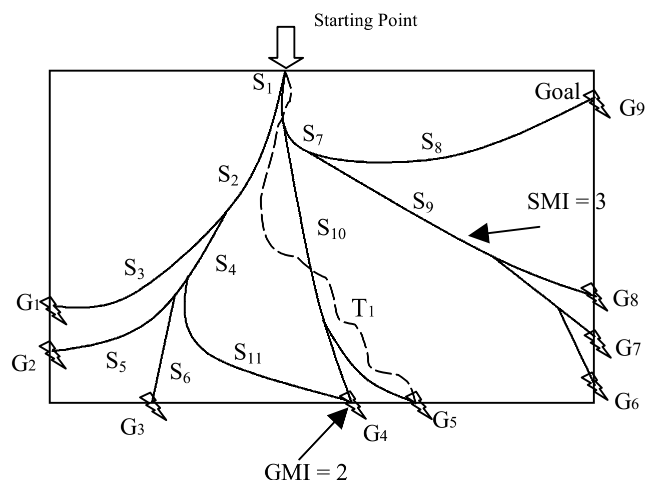

Figure 1: Terminology – S1 to S11 are strategies;G1 to G9 are goals; T1 is a sample trajectory; GMI is the Goal Multiplicity Index – how many strategies reach that goal; SMI = Strategy Multiplicity Index – how many goals this strategy lead to.

3.2 THE HYBRID MODEL

The hybrid model $\boldsymbol{M}$ for entity dynamics is of the following form:

$$M = p\chi + (1-p)\Gamma \quad (1)$$

where χ is any conventional dead reckoning model, Γ is one of several long-term entity strategy models and p is a binary weighting factor governed by:

$$p = 1 \text{ for } \|M - \Gamma\| \geq \theta$$
$$= 0 \text{ otherwise} \qquad (2)$$

where θ represents a distance measure threshold between the modeled behavior and the long term model Γ. In this way a model Γ based on *a priori* data can be employed when the entity movement is 'close' to such trajectories. In this paper we explore the idea that we may have several models from which to pick Γ based on *a priori* information. The particular model used is dynamically chosen so that improved fits can be obtained over single model approaches. At any instant in time the nearest model from the set available is used in equation (1).

Participating clients in a DIA use the model given by M. The parameters and initial entity state used by the model are updated every time the state deviates from the true state by a predefined threshold amount, $\boldsymbol{T_m}$.

While alternative soft model blending techniques could be employed here, we have opted for a simple switching technique between models to illustrate the principles involved. Section four describes in detail the strategy model, Γ.

4 CONSTRUCTING THE STRATEGY MODEL

4.1 THE TEST ENVIRONMENT

The long-term strategy models Γ employed in the hybrid prediction technique can be constructed in various ways: (a) by recording past actual entity movements in the environment and performing statistical analysis, (b) by heuristically identifying possible strategies based on the examination of the environment and (c) by employing automatic path-finding techniques. For this paper we chose the first method, developing a game-type Java application that records user trajectories in a controlled two-dimensional environment.

Users were asked to navigate from a fixed starting position to a fixed target position in as short a time as possible. Their view of any obstacles in their path was restricted to a circular area around their immediate position. The user began with no knowledge of the location of the target and repeated the exercise for a set number of trials. Each attempt constituted a trajectory. Data was collected and stored for each trajectory. This provided the basis of the statistical-based strategy models that we used in the hybrid contract technique. One of the two mazes employed is shown in Figure 2.

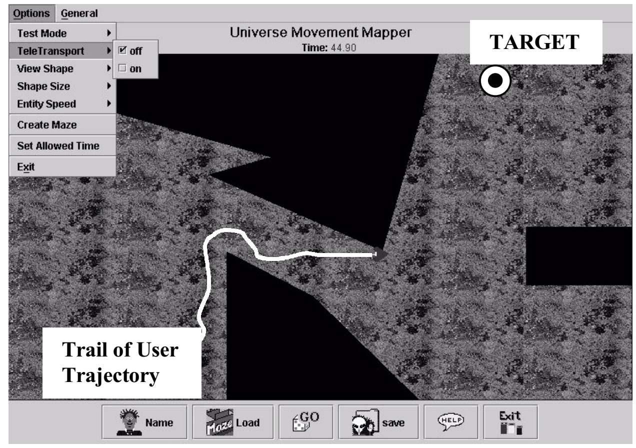

Figure 2: A screen shot of the trajectory recording software. The black areas are obstacles that do not allow users to pass.

4.2 CHOOSING THE STRATEGY MODEL

Using the software application, a minimum of five and maximum of fourteen trajectories were recorded from fourteen different users. The final trajectories for each user may be considered steady-state trajectories. Figure 3 plots these trajectories for ten users.

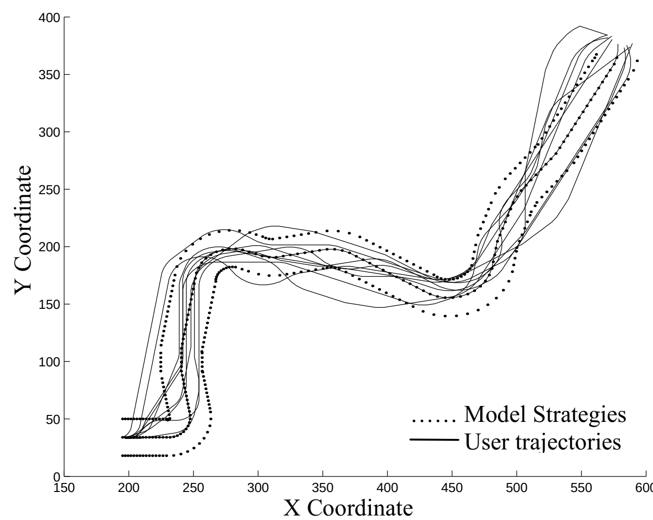

Figure 3: A plot of the final user trajectory for 10 different users. The three strategy models for a threshold of 10 are indicated as dotted lines.

From this plot we can identify a suitable steady-state strategy. In this paper we use a multiple strategy model to represent the given data. One user's steady-state strategy was chosen to represent the central strategy model. Two additional strategies were generated by spatially shifting the steady-state strategy by a fixed amount to generate three parallel strategies. The shifted strategies were created ased on the trigonometric considerations of Figure 4.

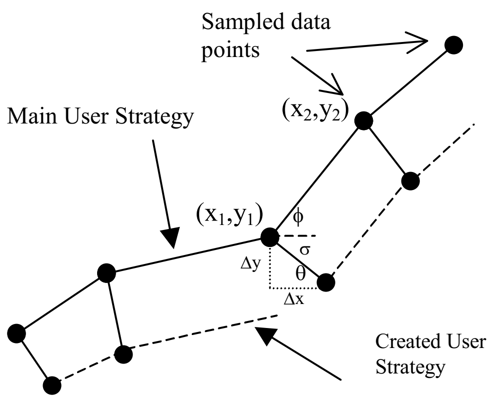

Figure 4: Diagram illustrating how the additional strategies are created. The solid line represents the steady state strategy and the dashed line represents one of the generated strategies.

The calculations are given in the following equations.

$$\phi = \tan^{-1}\frac{y_2 - y_1}{x_2 - x_1} \qquad \theta = \frac{\pi}{2} - \phi \tag{3}$$

$$\Delta x = \sigma Cos\,\theta \text{ and } \Delta y = \sigma Sin\,\theta \tag{4}$$

The (x_i,y_i) coordinates for the side strategies are given by: $(x_i+\Delta x_i,\ y_i-\Delta y_i)$ and $(x_i-\Delta x_i,\ y_i+\Delta y_i)$, where (x_i,y_i) are the coordinates of the central strategy. σ is chosen to be 1.6 times the threshold value, T_m. This ensures that when the entity lies within the area contained between the strategy models, dead reckoning will never be needed. It also provides a slight hysteresis, so that multiple switching will not occur on the boundaries between strategy coverage areas.

5. RESULTS

The results presented are based on two different environments similar to the one illustrated in Figure 2 above. These are given in sections 5.1 and 5.2 respectively. Both pure dead reckoning (DR) and our hybrid multiple-strategy (HMS) are considered. Results for a hybrid single-strategy (HSS) are presented for comparison purposes, where the single strategy is the central strategy from the HMS.

5.1 CASE STUDY 1

Of fourteen data sets recorded for this test case, ten were used to generate a strategy model. Of the remaining, four users were randomly chosen as representative test datasets.

Table 1 compares the number of packets sent for all trial trajectories for these two user datasets using the three models, first order DR, HSS and HMS. The threshold T_m was set at 25 in each case. Table 2 shows the results for the same techniques but this time with a threshold T_m of 10 in each case.

	Pure DR		**HSS**		**HMS**	
Trial	*User 1*	*User 2*	*User 1*	*User 2*	*User 1*	*User 2*
1	31	16	26	13	25	8
2	26	22	21	16	20	16
3	18	28	12	28	7	29
4	8	15	1	1	1	1
5	10	18	4	9	3	7
6	14	13	10	8	8	5
7	13	13	8	5	8	3
8	8	10	4	4	3	3

Table 1: The number of packets transmitted for two user sets for pure dead reckoning, the single strategy hybrid method and the multiple strategy hybrid method. Trial 1 is the initial trial. Threshold value: 25.

	Pure DR		**HSS**		**HMS**	
Trial	*User 1*	*User 2*	*User 1*	*User 2*	*User 1*	*User 2*
1	58	45	60	47	57	44
2	39	38	46	31	41	26
3	25	52	23	53	22	53
4	12	28	9	9	5	5
5	19	29	17	21	10	19
6	22	24	21	12	20	8
7	24	20	27	17	22	14
8	13	16	11	13	9	11

Table 2: The number of packets transmitted for two user sets for pure dead reckoning, the single strategy hybrid method and the multiple strategy hybrid method. Trial 1 is the initial trial. Threshold value: 10.

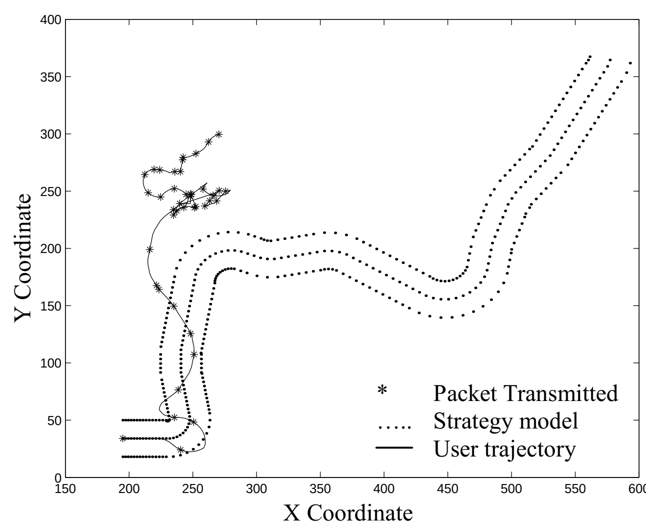

Figure 6a: A plot of the first trajectory attempt for user two. The user wanders, attempting to locate the target (to the top right). The threshold is set to 10.

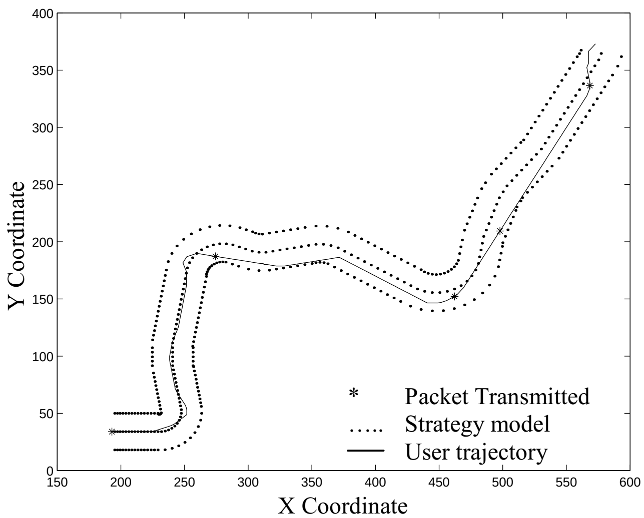

Figure 6b: A plot of the fourth trajectory attempt for user two. The user trajectory reaches the target. The strategy models are used without any dead reckoning. The threshold is set to 10.

Two trials for user 2 and a threshold of 10 are shown in Figures 6a and 6b. Each plot shows the strategy models, the user trajectory and the packets transmitted.

The initial trajectory of user two is shown in Figure 6a. This is a typical example of a transient strategy as the user wanders around the environment seeking the target. In this case the target is located at the top right end of the strategy model curve. Here DR is mostly used and the HMS method is rarely employed. This is to be expected, as the user has not yet formed a steady-state strategy.

In Figure 6b, trajectory four of user 2 is displayed. In this case the user has had three previous attempts and is more familiar with the environment. The trajectory is much more mature and purposeful and reflects a steady-state strategy. In such cases dead reckoning is rarely used.

For a threshold of 25 both hybrid methods perform better, in general, than pure dead reckoning, with the HMS technique giving the best results. Reducing the threshold value improves the remote entity-state fidelity but requires more packets to be transmitted in all three cases. Nevertheless the hybrid multiple-strategy technique still performs significantly better than pure dead reckoning.

5.2 CASE STUDY 2

Here a more complex environment was used to further illustrate the advantages of the HMS approach. Tables 3 and 4 show the results for each modeling technique for threshold values of 25 and 10 respectively.

These results illustrate that during initial attempts the various users do not adopt any of the chosen strategy models – see Figure 7a. Similar results were observed for case study 1. Once they have located the target within the environment they rapidly adopt a steady-state strategy that will allow them to minimize the time required to return to the target, as shown in Figure 7b. In some instances the users explore the maze to discover an alternative route or to better familiarize themselves with the environment– see for example user 1, trial 5.

Again, the results show that the HMS performs better than DR for both values of threshold.

	Pure DR		**HSS**		**HMS**	
Trial	*User 1*	*User 2*	*User 1*	*User 2*	*User 1*	*User 2*
1	52	35	44	42	34	30
2	25	38	23	32	18	24
3	34	22	23	6	19	5
4	21	18	4	3	3	3
5	24	16	20	6	11	5
6	18	20	7	3	5	3

Table 3: The number of packets transmitted for two user sets for pure dead reckoning, the single strategy hybrid method and the multiple strategy hybrid method. Trial 1 is the initial trial. Threshold value: 25.

	Pure DR		**HSS**		**HMS**	
Trial	*User 1*	*User 2*	*User 1*	*User 2*	*User 1*	*User 2*
1	78	54	83	64	80	62
2	41	55	46	56	46	54
3	57	32	56	35	49	24
4	32	26	26	24	18	11
5	45	26	40	14	38	12
6	31	26	22	34	15	21

Table 4: The number of packets transmitted for two user sets for pure dead reckoning, the single strategy hybrid method and the multiple strategy hybrid method. Trial 1 is the initial trial. Threshold value: 10.

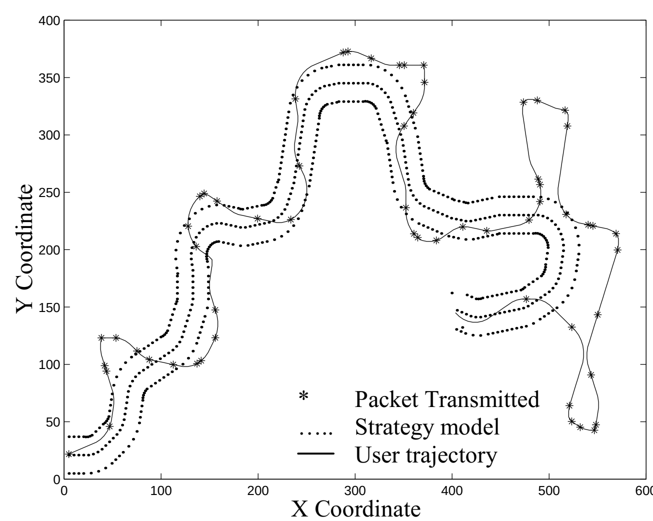

Figure 7a: A plot of the first trajectory attempt for user two. The trajectory wanders and explores different dead ends within the environment. The threshold is set to 10.

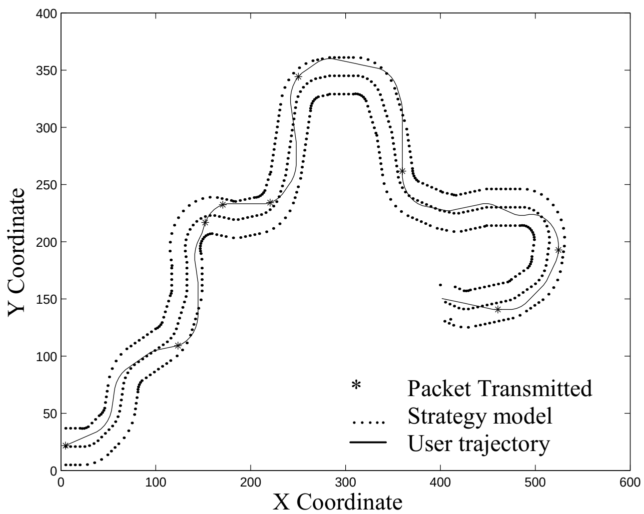

Figure 7b: A plot of the fourth trajectory attempt for user two. The trajectory is steady and progresses directly to the target, illustrating a steady state user strategy. The threshold is set to 10.

6. CONCLUSIONS AND FUTURE WORK

The results presented in this paper show that experienced users in a simple game-like environment with well-defined goals very quickly converge on common strategies to achieve them. By integrating this information into a novel hybrid multiple-strategy approach this paper has shown that there is a reduction in the number of transmission packets required to maintain fidelity across a Distributed Interactive Application.

Future work will involve looking at alternative techniques for extracting the long-term strategy models. In the case of novice users we are hoping that artificial intelligence techniques may help us better guess what they may do in particular environments. Model blending is also being investigated in order to achieve more natural convergence of entities when state-update information requires a correction to be made. Another area of research is to consider techniques for the quantification and assessment of the psycho-perceptual effects of these algorithms on users.

ACKNOWLEDGEMENT

This work was funded by Enterprise Ireland SC/2002/129/.

REFERENCES

[1] SINGHAL, S. K., AND ZYDA, M. 1999. *Networked Virtual Environments*. S. Spencer. ACM Press.

[2] DIOT, C. AND GAUTIER L. 1999. A Distributed Architecture for Multiplayer Interactive Applications on the Internet, *IEEE Network*, July/August 1999, 6-15

[3] FAISSTNAUER, C., SCHMALSTIEG, D., AND PURGATHOFER, W., Scheduling for very large Virtual Environments and Networked Games using Visibility and priorities. In *Proceedings of the 4th IEEE International Workshop on Distributed Simulation and Real-Time Applications,* San Francisco, California 24 - 26 August 2000, 31-38.

[4] LIANG, L.A.H., CAI, W., LEE, U. S. AND TURNER, S. J. 1999. Performance Analysis of Packet Bundling Techniques in DIS. In *Proceedings of the 3rd IEEE International Workshop on Distributed Interactive Simulation and Real-Time Applications*, Maryland 23 - 24 October 1999, 75-82.

[5] TANNENBAUM, A.S. 1996. *Computer Networks Third Edition*. Prentice-Hall.

[6] CAI, W., LEE F.B.S., AND CHEN L. 1999. An Auto-adaptive Dead Reckoning Algorithm for Distributed Interactive Simulation. In *Proceedings of the 13th Workshop on Parallel and Distributed Simulation*, Atlanta, 1 - 4 May 1999, 82-89.

[7] VALENTINO, G.J., THOMPSON, S.T., KNIOLA, T. AND CARLISLE, C.J. 1998. An SMP-based, Low-latency, Network Interface Unit and Latency measurement System: the SNAPpy system. In *Proceedings of the 2nd IEEE International Workshop on Distributed Interactive Simulation and Real-Time Applications*, July 1998, 62-70.

[8] BASSIOUNI, M.A., MING-HSING, C., LOPER, M., GARNSEY, M. AND WILLIAMS, J. 1997. Performance and reliability analysis of relevance filtering for scalable distributed interactive simulation. *ACM Transactions on modeling and computer simulation*, (7)3, July 1997, 293-331.

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

[10] D. DELANEY, T. WARD, S. MCLOONE, 2003, '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,* (ISBN 0-88986-341-5), February 10-13 2003.

[11] DURBACH, C. AND FOURNEAU, J.M. 1999. Performance evaluation of a dead reckoning mechanism. In *Proceedings of the 2nd IEEE International Workshop on Distributed Interactive Simulation and RealTime Applications,* Atlanta, Georgia, 1 - 4 May 1999, 23-29.

[12] PULLEN, J. M. 1999. Reliable Multicast network Transport for Distributed Virtual Simulation. In *Proceedings of the 3rd IEEE International Workshop on Distributed Interactive Simulation and Real-Time Applications,* College Park, Maryland, 23 - 24 October 1999, 59-66.

[13] WORTHINGTON, B. G. AND ROBERTS, D.J. 2000. Encapsulating Network Latency Compensators in VRML. In *Proceedings of VWSIM (Virtual Worlds and Simulation Conference) Int. Conf. SCS Western Multi-conference,* San Diego, USA, January 2000.

[14] ROBERTS, D.J. AND SHARKEY, P. M. 1997. Maximising Concurrency and Scalability in a Consistent , Causal, Distributed Virtual Reality System, whilst minimising the effect of Network Delays. In *Proceedings of the 6th IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, June 1997, 161-166.

[15] FAISSTNAUER, C., SCHMALSTIEG, D., AND PURGATHOFER, W., Priority Round-Robin Scheduling for very large Virtual Environments. In *Proceedings of the IEEE Virtual Reality 2000 Conference*, New Jersey, March 2000, 135-142.

[16] CHASSOT, C., LOZES A., GARCIA F., DIAZ M., DAIRAINE AND L., ROJAS L. 1999. QoS required by a DIS application in a new generation Internet. *IEEE Spring Simulation Interoperability Workshop*, Orlando, March 14-19, 1999.