

# A Message Update Model for a Large-scale Multi-user Networked Virtual Environment System

<sup>1</sup> Yang-Wai Chow, <sup>2</sup> Ronald Pose and Matthew Regan

<sup>1</sup> yang.wai.chow@csse.monash.edu.au

<sup>2</sup> ronald.pose@csse.monash.edu.au

*School of Computer Science and Software Engineering, Monash University, Clayton, 3800 Victoria, Australia.*

**Abstract**— Networked virtual environments attempt to create virtual worlds where multiple users, who might be scattered across the globe, can interact together within a shared environment. In the past, only expensive high-end systems had the capabilities necessary for the displaying, interfacing and processing of virtual environments. However, advances in the various aspects of technology required for the realization of networked virtual environments on end-user platforms, has made it possible for such virtual environments to be adequately represented on the standard PC. This in turn has led to a growing number of virtual environment users among the general public. With the increasing popularity and demand for very large-scale multi-user virtual environments, especially in the rapidly growing field of massive multiplayer online gaming, efficient communication models are necessary to reduce the amount of network traffic and bandwidth requirements in order to cater for the influx of users. This paper introduces a message update model which seeks to reduce the overall network communication costs of large-scale multi-user client/server networked virtual environment systems. The model presented in this paper is based on a priority rendering technique used for a Head Mounted Display (HMD) virtual reality graphics architecture known as the Address Recalculation Pipeline (ARP).

**Keywords**— Communication costs, message updates, networked virtual environments, online gaming.

## 1. INTRODUCTION

Networked Virtual Environments (NVE) are real-time simulations of synthetic virtual worlds, within which multiple users can interact with one another as well as with objects located within a user's individual surroundings. One of the main characteristics of NVEs is that multiple users immersed in the system, might be physically or geographically located very far apart from one another, but may still interact through a shared sense of time, space and presence within the shared virtual environment.



A networked virtual environment system combines a number of different aspects such as computer graphics, interaction devices, and network architectures. High-end virtual reality graphics display systems range from Head Mounted Displays (HMD) to CAVE systems, while interaction devices such as data gloves and magnetic sensors are used to detect and capture motion. NVEs have been developed for both academic and commercial applications in a variety of areas, ranging from educational and engineering applications, to military training simulations, entertainment and so on [1]. Some examples of early well known NVE systems include, SIMNET [2] used for military training, NPSNET [3] and DIVE [4] developed for academic research, and MASSIVE [5] a distributed virtual reality conferencing system, to name a few. In recent years, advancements in the various technologies have made it possible for virtual environments to be represented adequately on the standard PC, leading to the development of very large-scale virtual environments that are no longer limited to research laboratories. VRML is an example of a scene description language used for 3D Internet virtual worlds, which is publicly available to the common Internet user [6].

Recently, a successful area of networked virtual environments has been in the field of multiplayer online gaming [7], [8]. In these applications thousands of users around the world, navigate and interact together within very large virtual worlds. As the size of the virtual world becomes larger and number of users increases, so does the amount of network traffic and bandwidth requirements for the system. This is due to the fact that each additional participant introduces added shared states and extra levels of interaction to the virtual environment. Thus increasing the amount of shared data and consequently resulting in the sending and receiving of more information required for virtual world state updates [1].

The amount of network traffic also depends on the communication architecture adopted by the NVE system. There are two main communications architectures used in NVE systems, these are the peer-to-peer and client/server architectures. In peer-to-peer

architectures, each client is equal in status and sends information directly to other participating clients as appropriate. In the client/server setup, each client sends information to a centralized server responsible for passing on information to other clients as appropriate [9]. Both these communication strategies have their respective advantages and drawbacks. The peer-to-peer approach is bandwidth intensive but avoids placing loads on a centralized server and generally introduces lower network delays. On the other hand, the client/server approach might slow down message transfer but requires the transmission of fewer messages because the server can act as a centralized arbiter. This approach also allows for the use of multiple distributed servers to balance the overall processing load and to overcome the potential bottleneck limitations of such client/server systems.

With the popularity and demand for large-scale NVEs nowadays, especially in the area of online gaming, effective ways of minimizing network traffic are required to cater for this influx of users. This paper is concerned with the implementation of a client/server system that introduces a method to reduce the overall network communication costs of very large-scale virtual environments hosting a large number of users. This is done by allowing for different sections of a virtual world to be updated over the network at different update rates. The client/server system was chosen due to its multiple-server scalability, as well as its less bandwidth intensive nature as compared to the peer-to-peer network architecture.

The suitability of the model described in this paper depends heavily on the tolerable level of consistency-throughput tradeoff, deemed acceptable for a particular NVE application. The reason for this is because this model sacrifices accuracy of the virtual environment's shared state by reducing overall information exchange. While this might cater for the support of more users, it is therefore only suitable for applications that do not require perfect or tight concurrency. This system is suitable for use in applications such as Massive Multiplayer Online Role Playing Games (MMORPG), where thousands of users interact within a virtual world. Each user has a limited interaction range that is much smaller than their viewing range. Objects at a distance therefore do not have to be accurately represented, as the user will only be allowed to interact with objects or other users that are close by. This however, does not necessarily apply in the case of applications such as military training simulations where consistency and accuracy in the representation of the virtual environment between all the users is absolutely crucial.



## 2. MOTIVATION

A technique known as priority rendering [10] was devised for use in conjunction with an Address Recalculation Pipeline (ARP). The ARP is a specialized graphics display architecture designed for use in immersive Head Mounted Display (HMD) virtual reality systems [11]. The purpose of the ARP is to reduce the latency suffered by conventional HMD systems due to user head rotations. A characteristic fundamental to this system is that the scene encapsulating a user's head has to be pre-rendered onto display memory, before viewport mapping is performed. Priority rendering takes advantage of the fact that in general, upon slight user movements not everything in the scene encapsulating the user's head has to be re-rendered. Multiple display memories are therefore used in priority rendering for the purpose of reducing the overall rendering load.

In priority rendering, different objects are rendered onto separate display memories at different update rates, before being combined to form an image of the whole scene. Objects located closer to the user are required to be rendered at higher rates compared to objects further away from the user. Different virtual world objects could then be rendered at different update rates, whilst still maintaining the illusion of reality. In this manner, the implementation of priority rendering effectively reduced the overall rendering load of the ARP virtual reality system. A method of dividing the virtual world and grouping objects into regions was introduced to somewhat ease the management of virtual world objects for priority rendering [12].

The network communication approach introduced in this paper is based on the framework of region priority rendering, in that different sections of a virtual world can be updated at different update rates. It is applied here in the context of NVE message updates in order to reduce the overall information exchange over the network, rather than for the purposes of rendering. A more detailed description will be provided in the system framework section of this paper.

## 3. RELATED WORK

This section provides a background of related work as well as some issues involved in the designing networked virtual environment systems. The overview of related work focuses on attempts used by others to reduce network communication costs.

### 3.1 *Networking Issues.*

A number of network factors affect the usability of NVE systems. The way these issues are handled or

dealt with can have a great impact on a particular system's characteristics in terms of reliability, consistency and scalability. Some fundamental networking attributes and how they relate to network traffic and message efficiency in NVE systems, as described in [1], [8] are provided below.

- *Latency* – Refers to the length of time taken for packets from a source to arrive at its destination. Latency is primarily due to factors such as the medium of transport and network routing. Network latency cannot be eliminated entirely. Latency due to transmission medium and routing is beyond the control of the NVE developer. NVE developers however can try to cater for this by seeking to minimize the effect of network latency on user performance.
- *Bandwidth* – Is the rate of data transfer, in other words the amount of data transmitted or received per unit time. Available bandwidth capacity is determined by the hardware used to transmit the data, for example dial-up modems, Ethernet connections, etc. Other than hardware limitations, bandwidth is also dependent on how much, how large and how often messages are sent, as well as the number and distribution of users. In addition, different transmission techniques like broadcasting, multicasting or unicasting also affect bandwidth utilization.
- *Computational power* – As the amount of incoming network data increases, so does the computational load on a system required for the processing of network messages. For NVEs with a large number of participants and frequent message exchanges, a huge percentage of CPU time might be tied down with the processing of network messages, leaving less processing power for other computations.

These factors highlighted above, stresses the necessity for the design of NVE systems to incorporate efficient network communication models. These issues also emphasize the need to decrease the quantity of messages exchanged through the network, especially for systems with a large number of users, in order to reduce network communication costs. Network communication characteristics such as network traffic and bandwidth utilization are directly related to the amount of information sent over the network.

### 3.2 Spatial Characteristics.

In large virtual environments, not everything will be directly relevant to an individual user. A user's experience in the virtual world will only be affected by

objects or other users in his/her vicinity. This gives rise to certain spatial exploits that may be used to minimize data exchange over the network. A number of schemes have been implemented or proposed by other researchers regarding the use of such spatial exploits in order to improve the communication efficiency for various networking platforms.

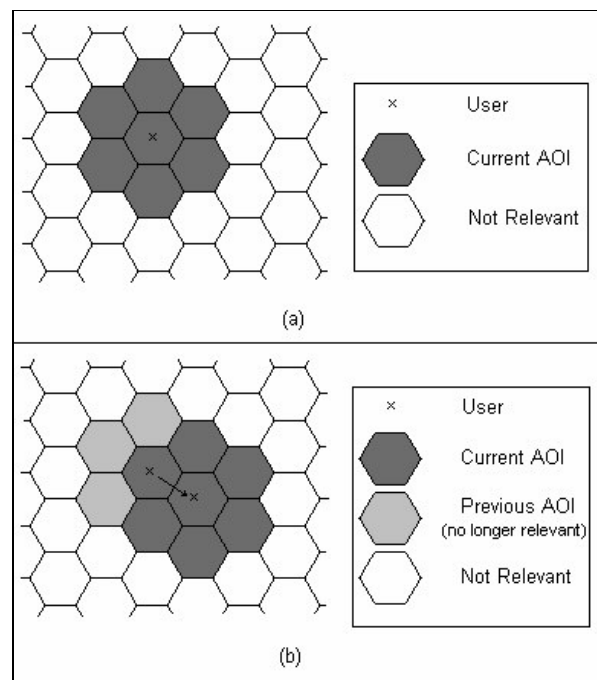


Figure 1: (a) Example of spatial AOI partitioning. (b) Updated AOI as user moves.

Some peer-to-peer NVE systems have employed certain network communication schemes that attempt to take advantage of a virtual environment's spatial characteristics, in order to minimize message passing as well as the number of required communication channels. Area of Interest (AOI) methods have been used to partition the virtual environment into a set of workable, smaller scale clusters where participants only establish communication channels and exchange messages among associated groups of interest, typically groups in the neighboring areas immediately surrounding the user [3], [13]. A relatively similar publisher/subscriber communications architecture has also been proposed for massive multiplayer games [14] by splitting the map into hexagon or square sections, where clients only subscribe to relevant sections of a virtual world. As a client moves through a virtual world the client will continually subscribe to certain sections and unsubscribe from sections that become no longer relevant. Figure 1a shows an illustration of the spatial partitioning of the virtual environment into AOI around a particular user. As the user navigates through the



virtual environment the AOI relevant to the user will change, as depicted in figure 1b.

The concept of locales used by Barrus, Waters and Anderson [15] is also based on the fact that in large virtual environments, objects or other participants that a user might observe at a given moment will only be local in nature. Locales seek to divide the world into smaller chunks, in order for them to be processed separately. This division of the virtual world into locales is by no means apparent to a user interacting within the system. Messages are only sent based on the neighboring locales surrounding the particular locale a user is currently located in. In this manner, locales irrelevant to the user are completely ignored.

Another method of network message management involves the culling of non-relevant information. The RING system [16] is a client/server system that was designed such that message updates were culled based on entity line-of-sight visibility. This meant that a client would only receive update messages containing information about the set of entities that the client could visibly see, no updates were sent regarding entities outside this viewing cone. This approach supported virtual worlds with dense occlusions and containing a large number of users, by effectively reducing the amount of information exchanged through the system. A space-scale structure approach was used in [17] for information culling in large-scale distributed virtual environments. In this approach, rejection functions were used to filter out irrelevant information based on a number of considerations, such as location and distance from an entity. The disadvantage of systems that implement such visibility or information culling is that a server has to compute for each and every individual client, the objects or other users in the virtual world that are visible and/or relevant from that particular client's point of view.

### 3.3 *Dead Reckoning.*

Dead reckoning is a technique used in distributed networked virtual environment systems in order to reduce the number of message communications required to maintain reasonable consistency between the participants. The aim is to reduce the update packet exchanges over the network by predicting the behavior or actions of virtual world entities. This also reduces the system's network dependency, thus improving the scalability of the NVE system by avoiding network bottlenecks [1], [18], [19].

Modeling the behavior of entities using dead reckoning algorithms has the effect of filling in the gaps between update packets, this allows for the smoother rendering of object movements. Dead reckoning therefore

accommodates for network latency experienced by the user from his/her perspective, in the sense that dead reckoning reduces the impact of network latency [20]. The drawback of dead reckoning is that the predicted behavior of an object might deviate significantly from the actual behavior. This however, will only be known and can only be corrected when the next update is received.

## 4. SYSTEM FRAMEWORK

A description of the system framework is given in this section, explaining the concepts and implementation of the message update model designed for large-scale multi-user NVE systems.

The idea for the framework was based on the region priority rendering concept, where a virtual world is divided into equal sized regions and different sections of the virtual world can be updated at different update rates depending on the proximity of particular regions to the user. Figure 2 depicts a 3D view showing the division of the virtual world into regions from a user's perspective. This concept is applied here for the purposes of implementing different rates, for the transmission of object status updates through the network from the server system to the individual clients.

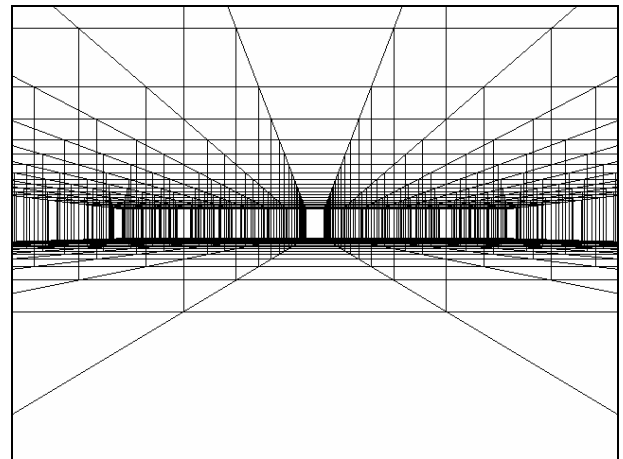


Figure 2: 3D view of regions from the perspective of a user.

The aim of this approach is to reduce the overall message update data sent by the server, whilst not affecting the general performance of individual users interacting within the shared virtual environment. Therefore in order to achieve this intended outcome, first and foremost the scope of a user's interaction range has to be pre-determined. Note that this interaction range is different from the user's viewing range. A user must be able to interact correctly and



accurately with the virtual world objects or other users within his/her vicinity without much delay. In this respect, it is therefore acceptable for the representation of objects located at a distance to be less accurate, because a user cannot immediately interact with these distant objects.

Methods have been used to cull or not represent distant objects. One of the various uses of fogging techniques in computer graphics is to hide the representation of far off objects, and to avoid distant objects from suddenly appearing or disappearing from the display as these objects enter into or exit from the user's viewing range. This however has the effect of cutting down the user's viewing range. A typical user will be less satisfied with this reduction in viewing range, as compared the inaccurate representation of distant objects. The reason for this is because even if the placement and actions of the objects at a distance might be inaccurate, by virtue of the fact that objects are still represented on the display creates the illusion of a much larger and wider viewing range. Thus giving rise to a more realistic and satisfying portrayal of the virtual environment. From the user's point of view (shown in figure 2), it can be seen that less accurate representation of objects in regions that are located at a distance further away from the user can potentially be tolerated, depending on the nature of the virtual environment application.

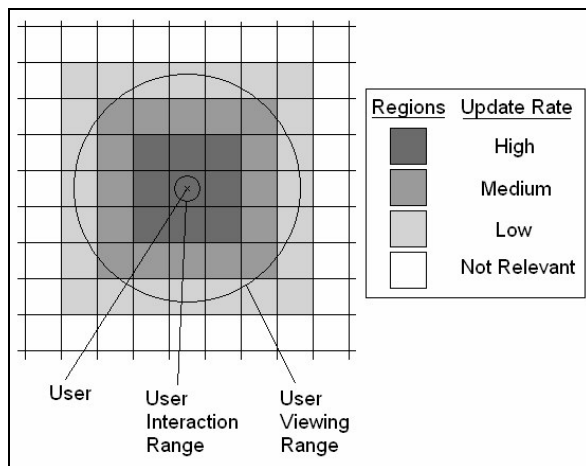


Figure 3: Top-down view of virtual world regions and their network update rates.

This gives rise to the possibility of dividing the virtual world into different regions, where the area surrounding the user can be updated at different update rates with little or negligible consequences to user performance and interactions. Figure 3 illustrates a top-down view of the division of the virtual world into square regions, the user's interaction and viewing range as well as the areas or regions in which different update rates can be

applied. As a user moves through the world and crosses region boundaries, the regions relevant to the user will change. Note that region sizes have to be larger than the user's interaction range. The appropriate size of the regions will have to be application specific, as the user's interaction range and viewing range will vary from application to application. This approach also allows a user to exercise flexibility in enhancing his/her viewing range, by simply communicating this request to the server, the server can then choose to send more region information updates to particular clients.

Other region shapes such as hexagons may also be used to partition the virtual world, as long as they cover the entire virtual world. It can be argued that dividing a virtual world into hexagonal regions reduces the number of contacts with neighboring regions and might subsequently decrease the amount of computational load required for the processing of surrounding regions. This is because a square region will be surrounded by eight neighboring regions, whereas a hexagon region will only be encompassed by six other regions, shown in figure 4. Nevertheless, square regions were used here for simplicity sake as well as for ease of use.

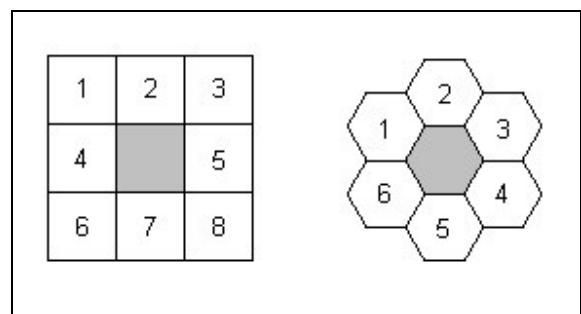


Figure 4: Number of neighboring regions for square and hexagon shaped regions.

In the implementation of this NVE system, each individual client interface is responsible for the displaying of the virtual world to the respective user. In addition, the client is also in charge of collecting input from the user regarding his/her actions and communications, before having to relay this information to the server. The server will therefore have knowledge about everything that happens in the virtual world and act as an arbiter for the virtual world by determining the validity of user actions, the handling of artificial intelligence of non-user entities, as well as for coordinating interactions between users and objects within the virtual world. By using the server as an arbiter, cheating by clients in the system can potentially be prevented. The server will also be responsible for sending update messages to the client systems



informing the clients about up-to-date events occurring around the user.

The message updating scheme used by the servers will be based on the region the user is located in and the surrounding regions. All users in the same region are to be sent the same update messages. This reduces the computational load involved in processing relevant message updates for each individual participant. Message updates for the various neighboring regions are performed at different update intervals as shown in figure 5 (refer to figure 3 for location of regions with respect to the user). Note that other update interval strategies are also possible, depending on the intended frequency of the updates. By adopting this message update scheme, not all objects located within the user's viewing radius have to be updated by the server through the network upon every single update cycle. Objects at a distance are allowed to be updated at lower update rates. The execution of the different update rates should decrease the amount of data exchanged over the network, therefore effectively reducing the overall network traffic and bandwidth utilization.

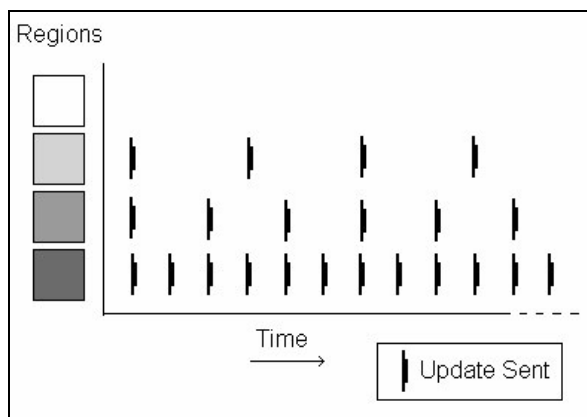


Figure 5: Illustration of the different region information update rates.

Dead reckoning techniques are to be used for predicting the behavior of objects, especially distant objects, in order to smoothen out the depiction of object movements. Even in the event that the predicted object behavior is completely wrong, the user will still need time to translate from his/her current position until the objects, which were previously only seen at a distance, are within interaction range. As the user gets closer to the region containing these objects, the server will be required to send updates for these objects at higher update rates. This will give time for the client system to correct the discrepancies between predicted and actual object behaviors by the time the objects are within the user's interaction range. In this manner, there should be

little or no ill effects on the user's interaction with these virtual world objects or other users.

## 5. EXPERIMENTAL RESULTS

Experimental simulations of a client/server networked virtual environment system implementing the message update model described above (referred to from here on as priority updating) were conducted. The constructed virtual world contained a large number of static and dynamic objects, as well as the emulation of other participants interacting together and with their surrounding environments. The goal of the experimental simulations was to verify the fact that by implementing the previously described priority updating system, whereby objects in different regions were to be updated at different update rates, for a client in the virtual world the average number of objects requiring updates would decrease by a significant amount.

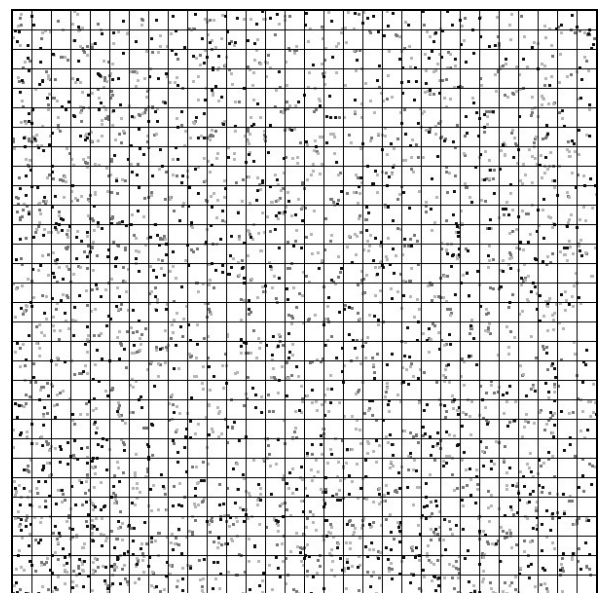


Figure 6: Status of the various virtual world entities coordinated by the server.

Figure 6 is a screenshot taken of the server's display at an arbitrary point in time during the simulation. It shows the status of virtual world's object locations and also the division of the entire world into square regions. Update messages concerning object behavior were sent to the clients based on these square regions. A screenshot taken from a client's display is shown in figure 8. This screenshot shows what a client sees of the virtual world from his/her point of view.



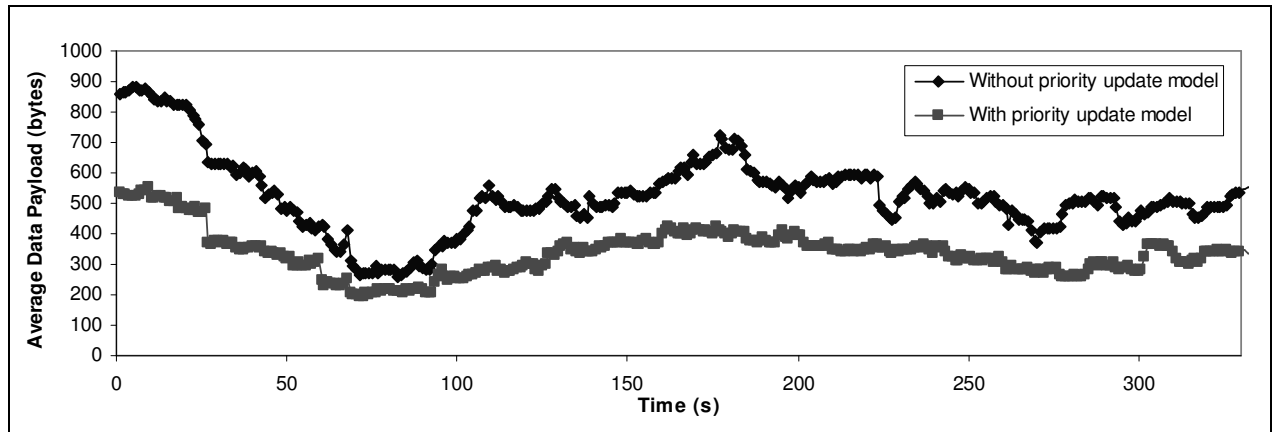


Figure 7: Average message update data payload.

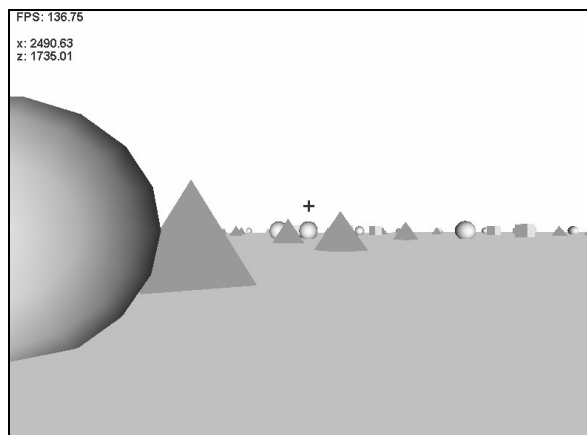


Figure 8: Screenshot of a client's display.

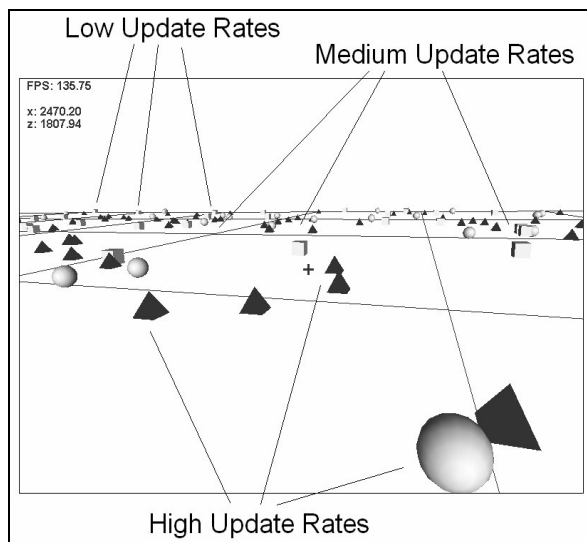


Figure 9: Screenshot showing region boundaries from an elevated perspective.

The screenshot in figure 9 shows the regions boundaries from an elevated user's perspective. Areas where the objects were updated at different update rates are highlighted in the picture. It can be seen that only objects at a distance from the user were allowed to be updated at lower update rates. Due to the fact that the user could not interact with these distant objects, inaccurate display of these objects would not affect the user's interaction with them. Moreover, the display of these distant objects are much smaller in size as compared to closer objects, therefore inaccuracies in distant objects' behaviors will be much less noticeable.

Two methods of message updating were used in the experiments. The first method involved updating all objects within a certain viewing radius around the user. The second method was the priority updating model, implementing the different region update rates. This was done in order to compare the efficiency of the update model in question. Measurements of the amount of information sent by the server to a client were taken as the user navigated through the virtual world. Figure 7 shows the average data payload of packets transmitted from the server. The data that was sent contained update information regarding the status of dynamic objects around that particular user. Each of the plotted payload values in figure 7 is an average of the data payload taken over a hundred update cycles. From the graph it can be seen that the overall update data sent over time is much less when using the priority update model. This is because not all objects had to be updated upon each and every update cycle.

Table 1 shows the average number of objects that were updated over the indicated period of time. These average values were again taken over a hundred update cycles. The reason why the average number of objects updated is significantly less for the case of priority



updating is because the objects in the different regions were update at different update rates. A closer and more detailed look at the exact number of objects updated using the different update rates, between the times of 180.544998 seconds and 181.466995 seconds, is depicted in figure 10. This figure shows that the number of objects that were updated changed periodically as a result of implementing the different update rates as previously illustrated in figure 5.

TABLE 1: AVERAGE NUMBER OF OBJECTS UPDATED.

| Time (s)   | Average Number of Objects Updated |                               |
|------------|-----------------------------------|-------------------------------|
|            | With Priority Update Model        | Without Priority Update Model |
| 174.998001 | 31.85                             | 53.47                         |
| 175.919998 | 33.52                             | 53.60                         |
| 176.841995 | 32.79                             | 58.46                         |
| 177.779999 | 31.94                             | 57.62                         |
| 178.701996 | 31.48                             | 55.00                         |
| 179.623001 | 30.93                             | 54.68                         |
| 180.544998 | 31.71                             | 54.66                         |
| 181.466995 | 32.51                             | 57.84                         |
| 182.389008 | 32.03                             | 57.31                         |
| 183.326996 | 31.92                             | 55.50                         |
| 184.248001 | 31.93                             | 53.16                         |
| 185.169998 | 30.17                             | 49.32                         |
| 186.091995 | 29.76                             | 48.64                         |
| 187.029999 | 29.75                             | 48.13                         |
| 187.951996 | 29.42                             | 46.47                         |

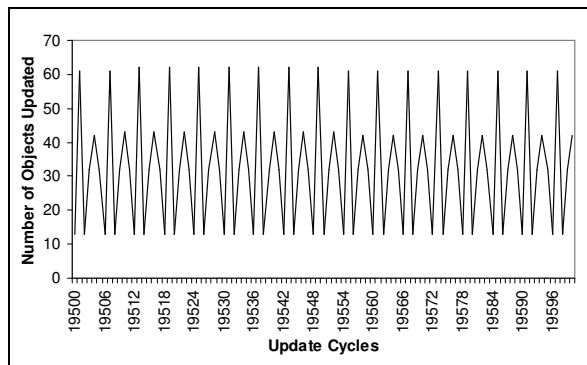


Figure 10: Number of objects updated per update cycle using the priority updating model.

This significant reduction in the amount of transmitted data is proportional to the reductions in the bandwidth utilization of the system, as less data is required to be transmitted per unit time. In NVE applications, the size of the data payload will depend on the amount of information required by individual applications for each object update, as well as whether or not data

compression techniques are used to compress the data. Packets sizes are also limited by Internet transmission protocols. Due to the fact that fewer objects require updating, this means that less data has to be sent. This in turn will affect the number of packets sent across the network, indirectly affecting the overall network traffic.

## 6. EVALUATION

The experiments performed were only small local area network simulations that did not have the complexity of a full-scale online NVE system. Even so, it was enough to establish the fact that the priority message updating model has the advantage of transmitting less overall information, whilst still maintaining usability and not greatly affecting the interactive nature of the virtual environment. This reduction in the required data exchange indicates that fewer and smaller packets are required to carry relevant shared virtual environment information across the network.

In light of the fact that less data is exchanged over the network, this also means that less processing power is required at the client side to process incoming network data. In this manner, more computational power of the client systems can be concentrated on other aspects of the virtual environment, such as dead reckoning, animation, rendering, etc. Dead reckoning techniques were not used in the simple experimental simulations. Nevertheless, it can be expected that the implementation of efficient and accurate dead reckoning techniques will improve the fluidity of object movement and translations as well as smoothen any animations in the display of the virtual world. This is especially important in a real online system where various network characteristics such as network latency, packet error or corruption as well as packet losses will have to be taken into consideration.

As the number of participants in a NVE system increases it becomes necessary to distribute and balance the computational load needed for maintaining and coordinating entities in the virtual world across multiple servers. Efficient algorithms for the distribution of computing resources among multiple servers are therefore essential in improving the scalability of the NVE system [21]. By adopting the region approach described in this paper, it is possible to spatially distribute the virtual environment across multiple servers, each handling different sections of the virtual world.

Figure 11 shows an example of region to server allocation. The figure shows the spatial partitioning of the virtual world into square-based regions. Spots depicted in figure 11 represent various individual



entities in the virtual environment, while the shaded areas show various regions managed by particular servers. As illustrated, the virtual world regions managed by the servers may differ in number. This is due to the fact that it is impractical to assume that users will be evenly scattered over the entire virtual world. Users will typically cluster around certain attractions located in the virtual world that will cause more users to frequent certain areas of the map more often than others. Therefore, load sharing techniques may be employed to distribute and share the computational load of the virtual world across the multiple servers. In this respect, some servers might potentially handle more regions than others. A dynamic load sharing algorithm for distributing square regions among multiple servers has been suggested by Ta and Zhou [22].

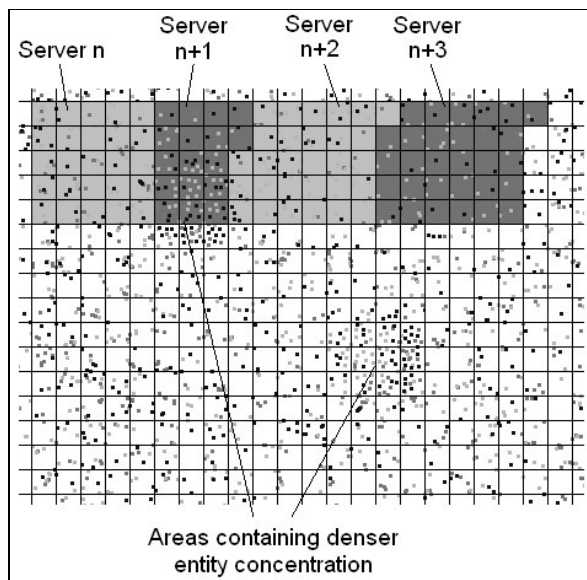


Figure 11: An example of spatially distributing the virtual world regions across multiple servers.

## 7. CONCLUDING REMARKS

The suitability of the message updating model presented in this paper relies heavily on the nature and genre of the intended networked virtual environment application. Its implementation will be advantageous for the purpose of reducing overall network communication costs such as network traffic and bandwidth utilization, by reducing the overall amount of information exchange. However this will only be applicable for NVE systems that can tolerate inaccurate representation of distant objects and where perfect consistency between the various users in the system is not essential. Online gaming applications such as Massively Multiplayer Online Role Playing Games (MMORPG) are examples of very large-scale NVE systems that can

tolerate inaccurate representation of distant objects and do not require perfect consistency for user interaction and satisfaction.

It should also be noted that the experiments conducted for this paper were carried out on a local area network, simulating a large number of dynamic entities. Due to practicality reasons, it would be extremely difficult to run real large-scale experiments over the Internet for a distributed networked virtual environment with thousands of users.

It is conceivable that this message update model can also be applied to a peer-to-peer network architecture, for the same purpose of reducing the overall message exchanges between all the virtual world participants. Future work will therefore involve modifying the message update approach presented in this paper, in order to accommodate the implementation of different update rates for a peer-to-peer networked virtual environment system.

## 8. REFERENCES

- [1] S. Singhal and M. Zyda, Networked Virtual Environments: Design and Implementation, NY: Addison-Wesley, 1999.
- [2] D.C. Miller and J.A. Thorpe, "SIMNET: The Advent of Simulator Networking," Proceedings of IEEE, vol. 83, no. 8, pp. 1114-1123, August 1995.
- [3] M.R. Macedonia, M.J. Zyda, D.R. Pratt, D.P. Brutzman and P.T. Barham, "Exploiting Reality with Multicast Groups: a Network Architecture for Large-Scale Virtual Environments," Proceedings of the Virtual Reality Annual International Symposium (IEEE VRAIS'95), Research Triangle Park, North Carolina, March 1995, pp. 2-10.
- [4] O. Hagsand, "Interactive Multiuser VEs in the DIVE System," IEEE Multimedia, vol. 3, no. 1, pp. 30-39, 1996.
- [5] C. Greenhalgh and S. Benford, "MASSIVE: A Distributed Virtual Reality System Incorporating Spatial Trading," Proceedings of the 15<sup>th</sup> International Conference on Distributed Computing Systems, Vancouver, British Columbia, Canada, June 1995, pp. 27-34.
- [6] B. Wolfgang, "Distributed Virtual Reality for Everyone – a Framework for Networked VR on the Internet," Proceedings of the Virtual Reality Annual International Symposium (IEEE VRAIS'97), Albuquerque, New Mexico, USA, March 1997, pp.121-128.
- [7] J. Kirkke, "South Korea Beats the World in Broadband Gaming," IEEE Multimedia, vol. 10, no. 2, pp. 12-14, June 2003.



- [8] J. Smed, T. Kaukoranta and H. Hakonen, "Aspects of Networking in Multiplayer Computer Games," The Electronic Library, vol. 20, no. 2, pp. 87-97, 2002.
- [9] S. Benford, C. Greenhalgh, T. Rodden and J. Pycok, "Collaborative Virtual Environments. To What Extent is Cyberspace Really a Space?" Communications of the ACM, vol. 44, no. 7, pp. 79-85. 2001.
- [10] M. Regan and R. Pose, "Priority Rendering with a Virtual Reality Address Recalculation Pipeline," Proceedings of SIGGRAPH 94, in Computer Graphics, Annual Conference Series 94, Orlando, Florida, United States, pp. 155-162, 1994.
- [11] M. Regan and R. Pose, "An Interactive Graphics Display Architecture," Proceedings of the Virtual Reality Annual International Symposium (IEEE VRAIS'93), Seattle, United States, September 1993, pp. 293-299.
- [12] Y.W. Chow, R. Pose and M. Regan, "Large Object Segmentation and Region Priority Rendering," To appear in Proceedings of the 28<sup>th</sup> Australasian Computer Science Conference (ACSC2005), Newcastle, NSW, Australia, 2005.
- [13] O. Hagsand, R. Lea and M. Stenius, "Using Spatial Techniques to Decrease Message Passing in a Distributed VE System," Proceedings of the 2<sup>nd</sup> Symposium on VRML, Monterey, California, United States, 1997, pp. 7-15.
- [14] S. Fiedler, M. Wallner and M. Weber, "A Communications Architecture for Massive Multiplayer Games," Proceedings of the 1<sup>st</sup> Workshop on Network and System Support for Games, NETGAMES 2002, Braunschweig, Germany, April 2002, pp. 14-22.
- [15] J.W. Barrus, R.C Waters and D.B. Anderson, "Locales: Supporting Large Multiuser Virtual Environments," IEEE Computer Graphics and Applications, vol. 16, no. 6, pp. 50-57, 1996.
- [16] T.A. Funkhouser, "RING: A Client-Server System for Multi-User Virtual Environments," Proceedings of SIGGRAPH 95, in Computer Graphics, Annual Conference Series 95, Monterey, California, United States, April 1995, pp. 85-92.
- [17] N.Farcet and P. Torguet, "Space-Scale Structure for Information Rejection in Large-scale Distributed Virtual Environments," Proceedings of the Virtual Reality Annual International Symposium (IEEE VRAIS'98), Atlanta, Georgia, United States, March 1998, pp. 276-283.
- [18] W. Cai, F.B.S. Lee and L. Chen, "An Auto-adaptive Dead Reckoning Algorithm for Distributed Interactive Simulation," Proceedings of the 13<sup>th</sup> Workshop on Parallel and Distributed Simulation, Atlanta, Georgia, United States, 1999, pp. 82-89.
- [19] T.P. Duncan and D. Gracanin, "Pre-reckoning Algorithm for Distributed Virtual Environment," Proceedings of the 2003 Winter Simulation Conference, vol. 2, pp. 1086-1093, December 2003.
- [20] L. Pantel and L.C. Wolf, "On the Suitability of Dead Reckoning Schemes for Games," Proceedings of the 1<sup>st</sup> Workshop on Network and System Support for Games, NETGAMES 2002, Braunschweig, Germany, April 2002, pp. 79-84.
- [21] J.C.S. Lui and M.F. Chan, "An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems," IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 3, pp. 193-211, March 2002.
- [22] N.B.D. Ta and S. Zhou, "A Dynamic Load Sharing Algorithm for Massively Multiplayer Online Games," The 11<sup>th</sup> IEEE International Conference on Networks (ICON2003), Sydney, Australia, Sept 2003, pp.131-136.

#### BIOGRAPHY

YANG-WAI CHOW received his B.Sc. and B.Eng. (Hons.) from Monash University, Australia. He has conducted Human Computer Interaction (HCI) research on web accessibility for the blind and vision impaired with the Human Factors Research Group at Telstra Research Laboratories. He has also been involved in research on streaming video traffic generation at the Centre of Telecommunications and Information Engineering (CTIE), Monash University. He is currently pursuing a Ph.D. in the area of virtual reality at the School of Computer Science and Software Engineering, Monash University, under the supervision of Ronald Pose and Matthew Regan.

RONALD POSE completed his B.Sc. (Hons.) and Ph.D. at Monash University, Australia. In 1987 he was employed as a Research Scientist at Telecom Australia Research Laboratories where he worked on the application of public key cryptography and certification techniques. He joined the faculty of Information Technology at Monash University in 1988. There he has supervised a number of research students with whom he has worked on a wide variety of research projects including neural networks, genetic algorithm function optimization, network routing, low latency virtual reality address recalculation pipeline display system, and self-reconfigurable computer systems. He has been involved with the Australasian Computer Architecture Conference (ACAC), now called Asia-Pacific Computer Systems Architecture Conference (ACSAC), since its inception.



MATTHEW REGAN received his B.Sc., B.Comp. (Hons.) and Ph.D. degrees in computer science at Monash University, Australia. His award winning Ph.D. on the Address Recalculation Pipeline (ARP) and Priority Rendering (U.S. patent) was supervised by Ronald Pose. After his Ph.D. he became a research fellow at Monash University and a research associate at Stanford University. He then joined Interval Research Corporation as a member of the research staff where he conducted research into low latency virtual reality. He is currently a member of the architecture team at nVidia Corporation in Santa Clara, California, United States.

