ICAT '99

Improving Scalability Using Sub-Regions in Distributed Virtual Environments

Mingyu Lim, Dongman Lee

Collaborative Distributed Systems & Networks Laboratory,
Information & Communications University
58-4 Hwaam-dong, Yusung-ku, Taejon 305-348, Korea
{cats, dlee}@icu.ac.kr

Abstract

As a distributed virtual environment (DVE) grows in size, a key aspect to consider is scalability. One of approaches to enabling a DVE system scalable is a region partition. Most DVE systems have regions disjoint and allow interactions between users in regions adjacent to each other. However, users of these systems have to be always informed of the status of all the users in neighbor regions that they're interested in. This imposes communications overhead on the users who wish to pursue interactions with other users across regions and thus makes the system less scalable. In our scheme, the region manager selects only a subset of users from the neighbor region whose members have high possibility of interaction with users in the current region. This enables users in the region not to receive all the update messages from the neighbor region. Our scheme allows DVE systems to support, in a scalable manner, inter-region interactions as well as intra-region interactions.

Key words: Scalable Region Management, Inter-Region Interaction, Large Scale Virtual Environments, Region Partition, Sub-Regions

1. Introduction

As a distributed virtual environment (DVE) grows in terms of number of users and network latency, a key aspect to consider is scalability for interactive performance [11]. One of approaches to enable a DVE scalable is an area of interest management [13]: dividing a virtual world into several regions [1, 5, 6, 10] or localizing the area of interest of the participants [2, 5]. It reduces the number of users managed by the system and thus the number of messages exchanged in the system. Most DVE systems have regions disjoint and allow interactions between users in regions adjacent to each other. Especially, DVE applications like a virtual shopping mall require frequent inter-region interactions as well as intra-region interactions. While a few systems such as NPSNET [10,12], MASSIVE [2,4] and SPLINE [1,15] support inter-region interactions, users have to pay the price: they must be always informed of the status of all the users in neighbor regions some of whom they're not interested in. This imposes communications overhead on the users who wish to pursue interactions with other users across regions and thus makes the system less scalable.

It is our motivation that users would interact mostly with adjacent users, instead of users in distance, in the neighbor region(s). We propose a new region management scheme to handle interactions between users in the neighbor region in a more scalable manner. In our scheme, the region manager selects only a subset of users from the neighbor region whose members have high possibility of interaction with users in the current region. This subset of users forms another multicast group. This enables users in the region not to receive all the update messages from the neighbor region. They receive the update messages regarding only the users in whom they are interested in the neighbor region. Thus, our scheme allows the large scale DVE system to support, in a scalable manner, inter-region interactions as well as intra-region interactions.

This paper is organized as follows. Section 2 discusses the various approaches of the existing systems. In section 3 we introduce on our scalable region management. Section 4 describes analysis of our mechanism. Section 5 describes experimental results of our mechanism. The conclusion follows in section 6.

2. Related Works

In this section, we describe the existing approaches for a scalable region management in large-scale DVE systems. We broadly divide them into two categories. One is region partition, and the other is aura management [3]. NPSNET [10,12] and SPLINE [1,15] (extended to Open Community) belong to the former while DIVE [5] and MASSIVE [2,4] to the latter. In WAVE [6,9], both are applied.

NPSNET is originally developed for battle simulation and designed in order to support large-scale virtual environments, partitioning the world into many hexagonal cells. All participants in one cell are grouped as a multicast group. Because they are interested in their own cell and six neighbor cells, they can interact with other participants in neighbor cells. When the participants interact with other participants in neighbor cells, they must get information of all the participants in neighbor cells. They receive the information that they are not interested in. This degrades the scalability of the system.

SPLINE introduces the concept of locale, like the cell in NPSNET. One locale constructs a multicast group and has information on the neighbor locales. Participants in one locale can interact with those in the neighbor locales. However, they must receive all the update messages emitted from their neighbor locales to do this. Since users in one region may not be interested in all these messages, thus this imposes communication overhead on networks and participant systems and reduces scalability. DIVE and MASSIVE introduce the aura concept to support natural user interaction and to reduce the communication cost. Since the user interaction in these systems is more important, they also present focus and nimbus, and support several awareness levels. In DIVE, one virtual world is one region, and it uses a portal for movement from one world to another via the gateway. For that reason, the region is isolated in DIVE, and the other regions are not visible to the participants in one region. DIVE handles no inter-region interaction management.

MASSIVE-1 and MASSIVE-2 are designed for teleconferencing system, so user interactions between partitioned regions do not occur in such a static situation. However, MASSIVE-2 uses the region partition and supports inter-region interaction using third party objects [2]. MASSIVE-3 distributes worlds based on locales in SPLINE. MASSIVE also has the same problem as NPSNET and SPLINE.

WAVE supports inter- and intra-region interactions. The world is partitioned into several regions and an aura manager manages each region. Since the aura managers are structured hierarchically, the movement of the participant from one region to another is performed via a parent aura manager. But WAVE also overlooks the interaction between regions.

3. Scalable Inter-Region Management

3.1 Sub-Regions

For interactions between users in adjacent regions, each user must know periodically the status of other users in the neighbor region such as position information. While users have high interest in the region that they belong to, they have relatively low interest in their neighbor regions. We deal with the differences of interest based on the adjacency between users. That is, users are interested in the whole region that they belong to, but their interest in the neighbor regions is likely to become diminishing as the distance from them increases.

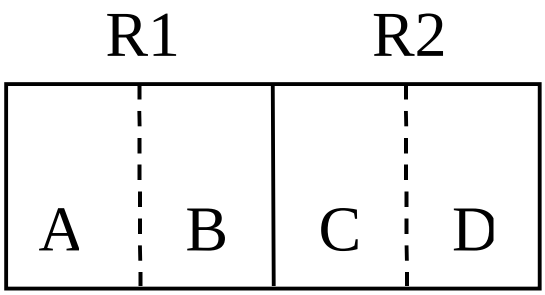

Fig. 1 Basic architecture of the inter-region interaction mechanism

As a user goes to the boundary of a region, it implies that the user becomes interested in the neighbor region. Therefore, it is unnecessary that the user always receives update information of all the users in the neighbor region. We assume that a user gets the world data of the neighbor region at a connection time. We divide regions into sub regions to support this.

Figure 1 shows two regions R1 and R2. We use these sub regions to distinguish the interest of users in the neighbor region. The sub region implies the relative difference of interest in the neighbor region. Users in a sub region that is near to a neighbor region have higher degree of interest in users in the neighbor region than those in a sub region that is distant from the neighbor regions. For example, users in B are likely to interact with users in the region R2 than those in A. Figure 2 represents the change of area which they are interested in. While the users in A get all data in R2, they only know update messages of users in C. The users in B get all update messages of users in C and D since they have more interest in R2 than those in A.

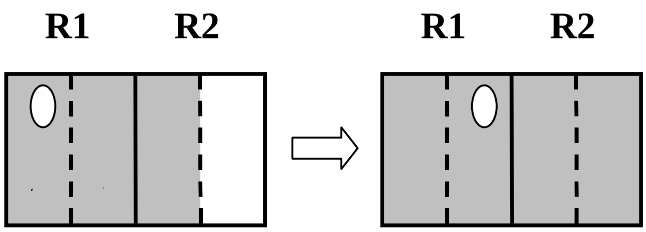

Fig. 2 Change of area which participants are interested in

3.2 Group Communication Model for Intra- and Inter-Region Interaction Management

The existing DVE systems assign a multicast address to each region by which users can interact with each other in the same region [1, 5, 6]. We distinguish interactions within a region and those with neighbor regions. To enable users to receive only subsets that they are likely interested in, we assign a separate multicast address to each sub region.

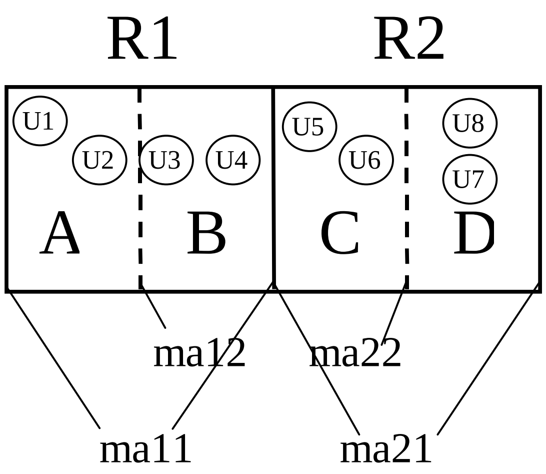

Fig. 3 Assignment of multicast addresses in regions

Since each user must join at least two multicast addresses, if they send messages using both addresses, they will receive duplicate messages that they have already received from one address. Users in an inter-region interaction group use a different multicast address according to whether they send or receive messages based on sub-region. A user multicasts update messages to other users who are interested in him using one multicast address. On the other hand, the user receives update messages multicasted from other users whom he is interested in using the other multicast address. The next sub section goes into details about assignment of multicast addresses.

Now we examine our group communication model for intra- and inter-region interactions in detail. In Figure 1, we can regard the sub regions B and C as boundary areas of region R1 and R2. To support inter-region interaction, we can assign an additional multicast address for a sub region. Figure 3 shows the assignment of multicast addresses in regions and users in each region. First of all, R1 and R2 need multicast addresses for intra-region interaction. From now on, we focus on inter-region interaction. The boundary areas B and C need additional multicast addresses for inter-region interaction of users who are interested in B and C, respectively. So we assign multicast address ma11 to R1, ma12 to B, ma21 to R2 and ma22 to C. ma12 and ma22 are assigned for only inter-region interactions. Since U1 and U2 in sub-region A are interested in sub-region C of R2, they join ma22 for inter-region interaction. Note that ma22 is only used when U1 and U2 receive update messages of U5 and U6. If U1 and U2 send their update messages with ma22, and if all users send or receive messages with their all multicast addresses, U5 and U6 receive duplicate update messages of U1 and U2 that they have already received from ma11. Since U7 and U8 in D are also interested in sub-region B of R1, they join ma12.

Since U3 and U4 in sub-region B of R1 are interested in R2, the whole region, they join ma21 to receive the update messages of all users in R2. U3 and U4 also use ma12 to send their update messages to U7 and U8 in sub-region D who are interested in users in B. Of course, they use ma11 to send their messages to U5 and U6 in sub-region C. In the same way, U5 and U6 in sub-region C join ma11 to receive update messages of all users in R1 and use ma22 to send their update messages to U1 and U2 in sub-region A. They use ma21 to send their update messages to U3 and U4. Using these additional multicast addresses, we reduce the number of messages exchanged for inter-region interaction. Table 1 shows a summary of what has been explained.

Table 1. Multicast addresses of intra- and inter-region interactions

	Intra-region Interaction	Inter-region interaction	
		send	Receive
U1, U2	ma11	ma11	Ma22
U3, U4	ma11	ma11, ma12	Ma21
U5, U6	ma21	ma21, ma22	Ma11
U7, U8	ma21	ma21	Ma12

We can extend our mechanism to a more general model. Until now, we have considered the virtual environment with two adjacent regions. Let's extend it to four adjacent regions as shown in Figure 4. Our interest here is inter-region interaction between R1 and R3 or between R2 and R4. The same mechanism is simply applied to these cases. We assume that users in a region cannot see the region in the diagonal location. As shown in Figure5, each region is divided into two sub-regions (vertical and horizontal). Additional multicast addresses are assigned for inter-region interaction to regions located vertically: ma13, ma23, ma33 and ma43. For inter-region interaction participants use different multicast addresses based on the mechanism explained above. As the number of neighbor regions increases, so does the number of multicast addresses to which users must join.

The vertical case is the same as the horizontal one. For example, U1 in sub-region A13 uses ma13 to send its update messages to U8 in sub-region C14 and use ma11 to send its update messages to U5 in sub-region C11. It uses ma31 to receive all update messages in R3. That is, U1 sends its update messages to users such as U8 in sub-regions C13 and C14 using ma13 and to users such as U5 in sub-regions C11 and C12 using ma11, while it receives messages from users such as U5 and U8 in region R3 using ma31. U1 also joins ma22 for inter-region interaction with R2.

U2 in sub-region A14 also uses ma11, ma13 and ma31 for its inter-region interaction with users in R3 like U1, and uses ma11, ma12 and ma21 for its inter-region interaction with users in R2. That is, while U1 and U2 join the same group for interaction with users in R3, they do to different groups for R2.

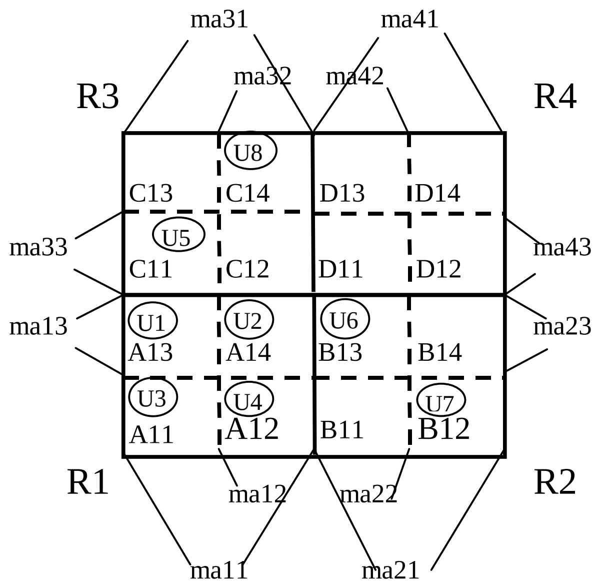

Fig. 4 Extension of scalable region management

Table 2. Multicast addresses of intra- and inter-region interactions for users in region R1

	Intra-region Interaction	Inter-region interaction			
		with R2		with R3	
		send	receive	send	receive
U1	ma11	ma11	ma22	ma13	ma31
U2	ma11	ma12	ma21	ma13	ma31
U3	ma11	ma11	ma22	ma11	ma33
U4	ma11	ma12	ma21	ma11	ma33

U3 and U4 also join the same group when they interact with users in R3, but join different groups when they interact with users in R2. When U3 interacts with users in R2, it belongs to the same group as U1. But when U4 interacts with users in R2, it belongs to the same group as U2. In the case of inter-region interaction with users in R3, U3 and U4 use ma11 and ma33. They multicast their update messages to users in C11 and C12 such as U5 using ma11. And they receive update messages of U5 using ma33.

Table 2 summarizes different addresses according to their interaction mode for users in region R1 in Figure 4.

4. Analysis

In this section, we analyze the scalability of our approach in terms of the number of messages exchanged for intra- and inter-region interactions, comparing with existing approaches such as Open Community. Suppose that the number of users in sub-regions A, B, C, and D in Figure 3 is m1, m2, n1 and n2, respectively. In the existing approaches, 2(m1+m2) is the total number of messages that users in region R1 needs to send since the users must send their update messages to both users in regions R1 and R2. That is, one is for intra-region interaction and the other for inter-region interaction. (m1+m2+n1+n2) is the total number of messages for each user in R1 to receive. Again, (m1+m2) is for intra-region interaction while (n1+n2) for inter-region interaction with R2.

On the other hand, when our scalable region management is used, the total number of messages for users in R1 to send is (m1+2m2). Users in sub-region A send their update messages once for intra-region interaction in R1 and for inter-region interaction with users in sub-region C. Users in sub-region B sends update messages to two multicast addresses. One is for intra-region interaction with users in R1 and the other is for inter-region interaction with users in sub-region D. The total number of messages for a user in R1 to receive is different depending on the sub-regions that the user belongs to. For example, (m1+m2+n1) is for users in sub-region A. Each user in A receives messages from (m1+m2) users in R1 for intra-region interaction and from n1 users in sub-region C for inter-region interaction. The total number of messages for a user in sub-region B to receive is (m1+m2+n1+n2). The user receives messages from (m1+m2) users in R1 for intra-region interaction and from (n1+n2) users in R2 for inter-region interaction.

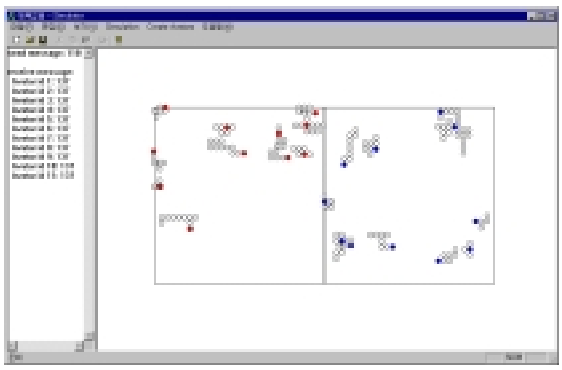

Fig. 5 Execution of the simulation

5. Experimental Results

Our scalable inter-region interaction mechanism is simulated as follows. We constructed a simplified virtual environment with two regions and implemented existing inter-region interaction mechanism and ours with MFC using Visual C++ 6.0. We ran the experiment on two Windows NT 4.0 connected by 10 Mbps Ethernet. On each machine runs the simulation program that simulates the state of one region. Each simulation program emulates one user's view. Because our focus is on inter-region interaction, we used IP multicast for inter-region interaction while intra-region interaction was just

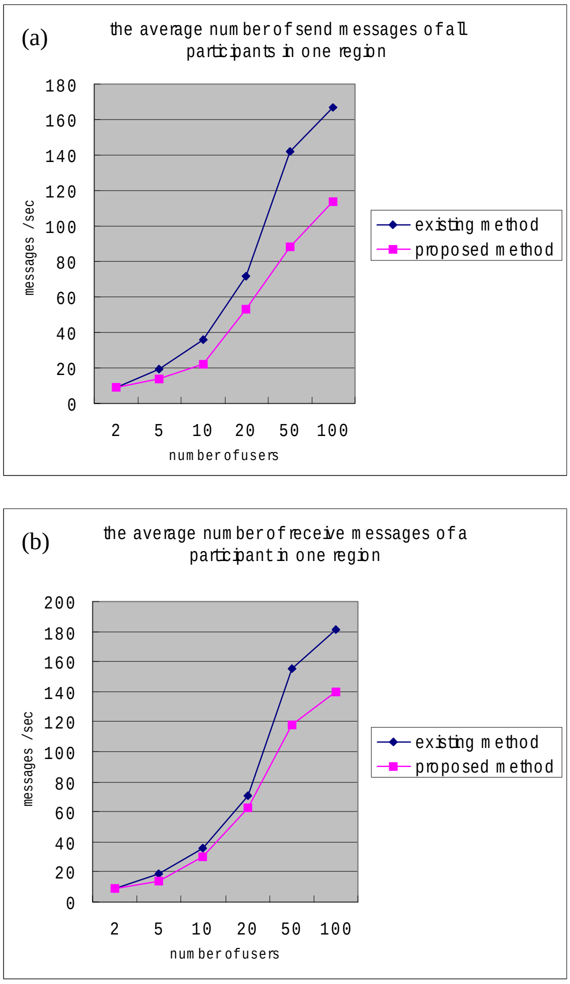

Fig. 6 Comparison of the number of messages between existing method and proposed method

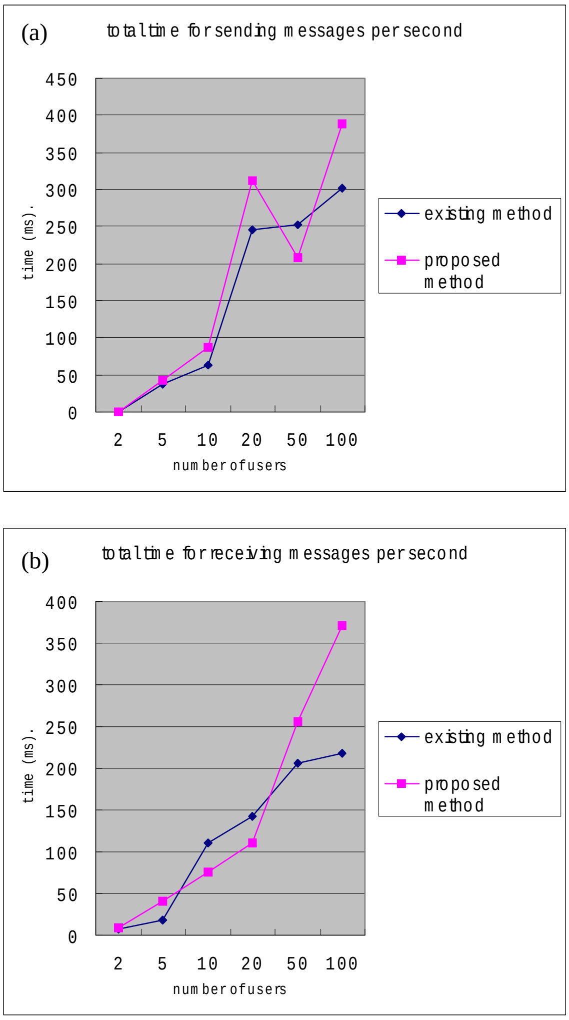

Fig. 7 Comparison of the local overhead between existing method and proposed method

simulated in each program. Users are simulated by instances of our user class. Each user is represented as simple filled circle. The users are created with random initial position and move in random direction. We assume that the speed of users is fixed with 10 pixels per 50 milliseconds. Figure 5 shows the execution of the simulation.

First, we evaluated the difference of the number of messages exchanged between general inter-region interaction mechanism and proposed mechanism according to increase of the number of participants in one region. Figure 6 shows the results. Figure 6(a) represents the average number of messages that all participants in one region sent per second. Figure 6(b) represents the average number of messages which one participant in one region received per second. Each value is the average of 100 times of iteration. As the number of participant increases, it shows that our scalable inter-region interaction mechanism requires much less messages than the existing mechanisms.

In the second simulation, we evaluated the overhead incurred by our inter-region interaction mechanism. While our mechanism reduces the number of messages, it may appear to come more local processing overhead than the existing methods. The overhead results from sub-region check and switching from one multicast address to another for inter-region interaction. Figure 7 shows the difference of local processing time of *total* update messages in one region. The range of the difference is from several tens of milliseconds to about 150 milliseconds, as shown in Figure 7. However, the overhead can be negligible. We assume that the frame rate is 20 frames/sec, which implies that the processing time per frame is 50 milliseconds. The difference of local processing time *per message* is at most several milliseconds regardless of the number of participants. The simulation locates the users randomly in each sub region and in some cases even the proposed approach has lower local processing time than the existing approaches. After all, there is no difference of local overhead between our approach and the existing approaches.

In summary, our approach imposes less message exchange overhead than the existing approaches.

6. Conclusion

We have proposed an enhanced region management technique that enables interactions with participants in the neighbor region in more scalable manner for large scale distributed virtual environments. The key idea is to divide a region into several sub regions, and assign additional multicast addresses according to the interest of the participants. It contributes to scalability by reducing communication overhead of inter-region interaction, which arises from fact that users do not have to be always informed of status of all the users in neighbor regions. We assume that users interact only with users whom they are interested in. So the scheme is of benefits to applications like large distributed virtual shopping mall, in which it is important for users to interact with others adjacent to them.

One may think that the allocation of additional multicast addresses for sub-regions costs but multicast addresses will not be expensive resources in the Internet over which IP version 5 is expected to be deployed widely [7]. The proposed scheme is part of our network framework for large scale distributed virtual environments, ATLAS [8]. The implementation of our scheme is underway onto CVRAT [14] system developed by KAIST. We currently investigate to find an optimal scope of boundary with simulation depending on the number of users in regions to reduce the communication cost as much as possible.

References

1. J.W. Barrus, R.C. Waters, and D.B. Anderson: “Locales: Supporting Large Multiuser Virtual Environments,” *IEEE Computer Graphics and Applications*, November, 1996, 16(6): pp. 50-57
2. S. Benford, and C. Greenhalgh: “Introducing Third Party Objects into the Spatial Model of Interaction,” *Fifth European Conference on Computer Supported Cooperative Work (ECSCW'97), Lancaster*, 8-10 September, 1997.
3. C.M. Greenhalgh, and S.D. Benford: “Boundaries, Awareness and Interaction in Collaborative Virtual Environments,” *in the proceedings of the 6th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE), Cambridge, Massachusetts, USA*, 18-20 June, 1997.
4. C.M. Greenhalgh, and S.D. Benford: “Virtual Reality Teleconferencing: Implementation and Experience,” *Proc. ECSCW'95, Stockholm, North-Holland*, September, 1995.
5. O. Hagsnad: “Interactive Multiuser VEs in the DIVE system,” *IEEE Multimedia*, Spring 1996, pp. 30-39.
6. O. Hagsand, R. Lea, and M. Stenius: “Using Spatial Techniques to Decrease Message Passing in a Distributed VE System,” *VRML 97, Monterey CA USA*, pp. 7-15.
7. D.C. Lee, D.L. Lough, S.F. Midkiff, N.J. Davis IV, and P.E. Benchoff: “The Next Generation of the Internet: Aspects of the Internet Protocol Version 6,” *IEEE Network*, 1998.
8. D. Lee et al: “ATLAS: Scalable network framework for large distributed virtual environments,” MIC *Project Report*, August, 1999.
9. R. Lea, Y. Honda, K. Matsuda, O. Hagsand, and M. Stenius: “Issues in the design of a scalable shared virtual environment for the internet,” *HICSS-30, Hawaii*, Jan, 1997.
10. M. Macedonia, M. Zyda, D. Pratt, D. Brutzman, and P. Barham: “Exploiting Reality with Multicast Groups,” *IEEE Computer Graphics & Applications*, September, 1995, pp. 38-45.
11. M.R. Macedonia, and M.J. Zyda: “A Taxonomy for Networked Virtual Environments,” *IEEE Multimedia, 4(1)*, Jan-Mar, 1997, pp. 48-56.
12. M.R. Macedonia, M.J. Zyda, D. Pratt, R. David, P.T. Barham, and S. Zeswitz: “NPSNET: A Network Software Architecture for Large Scale Virtual Environments,” *Presence, Vol. 3, No. 4*, Fall 1994, pp. 265-287.
13. S. Singhal, and M. Zyda: “Networked Virtual Environments Design and Implementation,” *Addison Wesley*, July, 1999.
14. J. Sung, J. Sim, and K. Wohn: “A heterogeneous multicast communication for the network virtual reality system,” *Korea Simulation Conference, Vol 7*, 1998, pp. 1-14.
15. R.C. Waters, D.B. Anderson, J.W. Barrus, D.C. Brogan, M.A. Casey, S.G. McKeown, T. Nitta, I.B. Sterns, and W.S. Yerazunis: “Diamond Park and Spline: A Social Virtual Reality System with 3D Animation, Spoken Interaction, and Runtime Modifiability,” *Technical report 96-02, MERL Cambridge MA*, January, 1996.