

# Providing Full Awareness to Distributed Virtual Environments Based on Peer-to-Peer Architectures<sup>\*</sup>

P. Morillo<sup>1</sup>, W.Moncho<sup>1</sup>, J.M. Orduña<sup>1</sup>, and J. Duato<sup>2</sup>

<sup>1</sup> Instituto de Robótica. Universidad de Valencia. Spain

<sup>2</sup> DISCA. Universidad Politécnica de Valencia. Spain

Pedro.Morillo@uv.es

**Abstract.** In recent years, large scale distributed virtual environments (DVEs) have become a major trend in distributed applications, mainly due to the enormous popularity of multiplayer online games in the entertainment industry. Since architectures based on networked servers seems to be not scalable enough to support massively multiplayer applications, peer-to-peer (P2P) architectures have been proposed as an efficient and truly scalable solution for this kind of systems. However, the main challenge of P2P architectures consists of providing each avatar with updated information about which other avatars are its neighbors. We have denoted this problem as the awareness problem. Although some proposals have been made, none of them provide total awareness to avatars under any situation.

This paper presents a new awareness method based on unicast communication that is capable of providing awareness to 100% of avatars, regardless of both their location and their movement pattern in the virtual world. Therefore, it allows large scale DVEs based on P2P architectures to properly scale with the number of users while fully providing awareness to all of them.

## 1 Introduction

In recent years, large scale distributed virtual environments (DVEs) have become a major trend in distributed applications, mainly due to the enormous popularity of multiplayer online games in the entertainment industry. These highly interactive systems simulate a 3-D virtual world where multiple users share the same scenario. Each user is represented in the shared virtual environment by an entity called *avatar*, whose state is controlled by the user through a client computer. The system renders the images of the virtual world that each user would see if he was located at that point in the virtual environment. Hundreds and even thousands of client computers can be simultaneously connected to the DVE through different networks, and even through Internet. DVE systems are currently used in many different applications [30], such as civil and military distributed training [20], collaborative design [29] and e-learning [4]. Nevertheless,

---

<sup>\*</sup> This paper is supported by the Spanish MEC under Grant TIC2003-08154-C06-04.

the most extended example of DVE systems are commercial, multiplayer online game (MMOG) environments [5, 14, 27, 2, 33].

Architectures based on networked servers have been during last years the major standard for DVE systems [30, 16, 35, 17, 9]. In these architectures, the control of the simulation relies on several interconnected servers. Client computers are assigned to one of the servers in the system. In these architectures, each new avatar represents an increase not only in the computational requirements of the application but also in the amount of network traffic. Due to this increase, networked-server architectures do not properly scale with the number of existing users, particularly for the case of MMOGs [1].

Although techniques like Frontier Sets [34] have been proposed for structured environments and small scale online games, these solutions cannot be extrapolated to massively multiplayer online games. The most adequate scheme in order to provide good scalability for large scale DVE systems seems to be P2P architectures, and several online games based on P2P architectures have been designed [22, 21, 8]. Nevertheless, P2P architectures must face *awareness problem*. This problem consists of ensuring that each avatar is aware of all the avatars in its neighborhood [32]. Solving the awareness problem is a necessary condition to provide a consistent view of the environment to each participant. Effectively, if two neighbors avatars are not aware of such neighborhood, they will not exchange messages about their movements and/or changes, and therefore they will not have the same vision of the shared environment. Thus, providing awareness to all the avatars is a necessary condition to provide consistency (as defined in [36, 7, 28, 31]). However, it is not a sufficient condition. Even when using a awareness method that determines at each moment which other avatars must each avatar exchange messages with, time-space inconsistencies can arise among different avatars because of clock drifts and/or network delays [36]. Awareness is crucial for MMOGs, since otherwise abnormal situations could happen. For example, a user provided with a non-coherent view of the virtual world could be shooting something that he can see although it is not actually there. Also, it could happen that an avatar not provided with a coherent view is killed by another avatar that it cannot see.

In networked-server architectures, the awareness problem is easily solved by the existing servers, since they know the location of all avatars during all the time. Each avatar reports about its movement (by sending a message) to the server where it is assigned to, and the server can easily decide which avatars should be the destinations of that message by using a criterion of distance. There is no need for a method to determine the neighborhood of avatars, since servers can easily do this task.

However, in DVE systems based on P2P architectures the neighborhood attribute must be determined in a distributed manner, in such a way that awareness is provided to all avatars during all the time. Currently, several strategies for providing awareness in DVE systems based on P2P architectures have been proposed [11, 13, 18, 12, 10, 8]. Unfortunately, some of this proposals [13, 18, 8] are based on multicast communications, therefore being unsuitable for MMOGs,

the most extended type of large scale DVE systems. The methods proposed in [11] and [12] do not guarantee awareness to all avatars (they do not provide a awareness rate of 100 %), since these methods do not guarantee proper neighbor discovery in all the possible cases [10]. Finally, the method proposed in [10] seems to provide full awareness. However, this proposal has not been evaluated neither on a real system nor with a simulation tool. The high number of neighbors that each avatar needs to communicate with in order to provide awareness suggests that this method is not able to provide full awareness in a scalable way.

In this paper, we propose a method that provides full awareness to large scale DVE systems based on peer-to-peer architectures in an actually scalable way. Performance evaluation results show that the proposed algorithm can provide full awareness in a large scale DVE system, even when avatars follow non-uniform movement patterns and they are unevenly distributed in the virtual world. Therefore, this algorithm can allow P2P architecture to become an actually efficient solution for large scale DVE systems like massively MMOGs.

The rest of the paper is organized as follows: Section 2 analyzes the existing proposals for providing awareness in DVE systems based on P2P architectures. Section 3 describes the proposed algorithm and how it improves the weaknesses of the existing proposals. Next, Section 4 presents the performance evaluation of the proposed method. Finally, Section 5 presents some concluding remarks and future work to be done.

## 2 Background

Peer-to-peer architectures were proposed some years ago for DVE systems [6]. However, during last years most of DVE systems have been designed following a networked-server architecture, and a lot of research has been targeted to improve the performance of DVE systems with this architecture [30, 25, 15, 3]. Nevertheless, the expansion of MMOGs has made large scale DVE systems to become usual, and networked-server architectures seem to lack scalability to properly manage the current number of avatars that these system can support (up to some hundred thousands of avatars [5]). As a result, some studies have proposed again the use of P2P architectures [11, 13, 18, 10, 12], since this schemes seems to be the most scalable one. Nevertheless, before a P2P architecture can be used to efficiently support large scale DVE systems, the awareness problem must be still solved.

Some proposals use multicast communications to guarantee awareness [13, 18]. Although multicast greatly improves scalability, it is hardly available on the Internet, which is the natural environment for multiplayer online games. Therefore, this scheme cannot be used in most of large-scale DVE systems.

The solutions proposed in [11] and in [12] use unicast communication, but they do not provide total awareness. For example, the method proposed in [11] is capable of providing a awareness rate of 95%. However, a awareness rate of 100% is crucial for MMOGs, since it guarantees that no faults like being killed by an "invisible" avatar will occur. The reasons that keep this scheme from providing

full awareness are on one hand that it limits the number of neighbors that a given avatar can see. When avatars move following a non-uniform movement pattern, some regions of the virtual world can be crowded. In this case this scheme cannot guarantee awareness. On the other hand, in this proposal the awareness of a given avatar  $i$  depends on the spatial location of the neighbors of avatar  $i$ . If the location of these neighbors is not uniform around avatar  $i$ , some other avatars could approach  $i$  without being detected by the neighbors of  $i$ . This case is particularly frequent when avatars follow non-uniform movement patterns. Something similar could happen if the method proposed in [12] is used, as described in [10].

A different approach uses Frontier Sets to provide awareness in DVE systems simulating a structured virtual world [34]. Nevertheless, in this approach awareness is based on exchanging information between each pair of avatars. Those avatars not having any frontier between them must exchange information about their location and actions 10 times per second, while those avatars having a frontier between them must check this frontier every 5-10 seconds. This massive exchange of information is very costly in terms of scalability. As a result, this proposal does not show that full awareness is guaranteed while maintaining scalability.

Finally, the method proposed in [10] is based on the use of Voronoi diagrams, and it seems to provide full awareness. However, the use of Voronoi diagrams makes this method require each avatar  $i$  to communicate with a high number of avatars, even though they are located far away from  $i$  and they have a small Area Of Interest (AOI) [30]. This feature suggests that this method is not able to provide full awareness in a scalable way. In fact, this proposal lacks a performance evaluation (either on a real system or with a simulation tool) that shows an actual scalability of the method while providing a 100% of awareness. Therefore, the efficiency and scalability of this proposal cannot be stated.

### 3 A New Awareness Method: COVER

In order to provide awareness to a given avatar  $i$ , the propose method (called COVER) involves all the avatars surrounding  $i$  up to the second level of neighborhood, like the method proposed in [10]. The first-level neighbors of an avatar  $i$  are those avatars in the DVE system in whose AOI avatar  $i$  appears. The second-level neighbors of  $i$  are all the neighbor avatars of the first-level neighbors of  $i$ . In order to avoid cyclic relationships (redundant messages), if a second-level neighbor is also a first-level neighbor, then it is not considered as a second-level neighbor. Each time an avatar  $i$  moves, it sends an updating message to each of its first-level neighbors. These neighbors in turn propagate the updating message of  $i$  to the second-level neighbors of  $i$ .

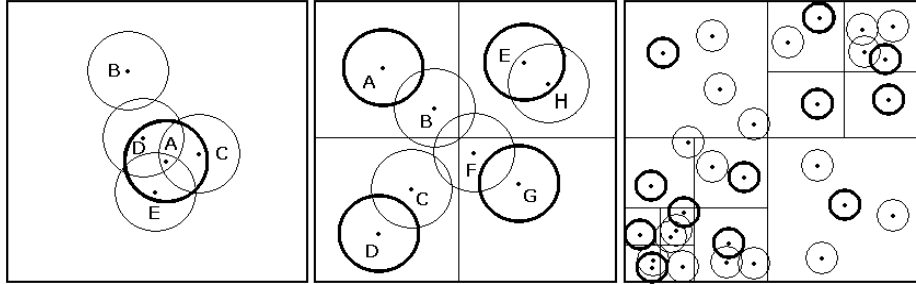
Unlike the methods proposed in the previous section [11, 10], COVER classifies avatars in two categories: covered or uncovered. Each avatar checks its classification each time it moves and each time that any of its neighboring avatars moves. We denote an avatar  $i$  as *covered* if its first or second-level neighbors are located in such a way that the intersections of their AOIs totally cover the

AOI of  $i$ . Otherwise, it is considered as an *uncovered* avatar. COVER method offers auto-awareness to covered avatars, because no avatar can approach them without being detected by their neighbor avatars.

Unlike the method proposed in [11], our approach provides awareness also for uncovered avatars, by means of using *supernode* avatars. These avatars play multiple roles, acting not only as simple avatars but also as pseudo-servers [30]. Supernodes represent an upper layer in the awareness scheme, and they provide the required scalability while ensuring a awareness rate of 100%. Supernodes are responsible of providing awareness to the uncovered avatars in their surroundings, and they are initially designated by the entity in charge of the initialization of new avatars (denoted as *Loader* [26] or Bootstrap server [11]) when they join the DVE system. At boot time, the loader divides the 3-D virtual scene into square sections called *regions*. For each region, the closest avatar to the geometric center of each region is selected by the loader as the supernode for that region. From that instant, supernodes are responsible of providing awareness to those uncovered avatars that are located within their regions. Uncovered avatars must send their updating messages not only to their neighbors, but also to the corresponding supernode of the region where they are located. In this way, supernodes can notice uncovered avatars when another uncovered avatar(s) cross their AOI. The auto-awareness of covered avatars ensures that before a covered avatar  $k$  can enter the AOI of an uncovered avatar  $i$ , another uncovered avatar  $j$  will cross the AOI of  $i$ . Therefore, this scheme does not require supernodes to notice uncovered avatars about the movement of covered avatars, significantly reducing the communications required for providing awareness.

As an example, figure 1-left shows a 2-D region containing five avatars, represented as dots, and their respective AOIs, represented as circumferences around the dots. In this region avatars B, C, D and E are uncovered avatars. Since the circumference around avatar A is totally covered by the AOIs of avatars D, C and E, avatar A is classified as a covered avatar. Also, in this case avatar A has been chosen as supernode of the region (we have represented supernodes by depicting their AOI with a thicker circumference), and therefore this avatar will receive updating messages from all the uncovered avatars in this region (the rest of the avatars).

COVER limits the maximum number of uncovered avatars which are simultaneously connected to the same supernode. This parameter is called *MNUA*, (for Maximum Number of Uncovered Avatars). Whenever MNUA is exceeded, the supernode divides that region in four different subregions and computes a new supernode for each subregion, based on the criterion of geometric distance to the center of the subregion. Once the division has been performed and a new supernode is selected for each subregion, the uncovered avatars in each subregion are re-assigned to the new supernodes. It is worth mention that the criterion used for selecting new supernodes does not distinguishes between covered or uncovered avatars. When the system is running, this mechanism defines a dynamic quad-tree structure where each supernode has four sons. Two or more supernodes are *brothers* if they have been generated in the same division operation. Brother



**Fig. 1.** An example of a virtual scene composed of a single region where a covered avatar is the supernode at the same time (left). As more avatars come into the scene a division into four (center) or more regions (right) can be performed.

nodes are constantly monitoring each other to detect if the total number of uncovered avatars located in the four regions is under MNUA. In this case, a fusion operation is performed. In this operation the four brother subregions are joined to become a unique, larger region, and a new supernode for the new region is computed based on the same criterion of distance to the geometric center of the resulting region.

As an example, figure 1-center shows the evolution of figure 1-left when the proposed scheme is applied and the MNUA parameter has value of six avatars. Since three more avatars have joined the system and the MNUA value has been exceeded, the supernode has divided the region in four subregions. In this case, the resulting supernodes are now avatars A, D, E, and G. These brother supernodes monitor the total number of uncovered avatars in the zones that they control. COVER method does not require that supernodes exchange the position of the avatars in their respective regions, like networked-server architectures do [30, 3]. This key issue allows COVER method to limit the amount of messages generated to provide awareness.

In order to offer full awareness to those avatars located at the borders of different regions (denoted as *critical* avatars), *secondary* supernodes are used. Critical avatars are defined as those uncovered avatars whose AOIs intersect with more than a single region. Critical avatars should send updating messages not only to the supernode managing the region where they are located, but also to the supernodes managing the adjacent regions. These supernodes show uncovered avatars located in different regions, and they are considered as secondary supernodes for critical avatars. Figure 1-center shows that avatar B and F must be considered as critical avatars, because the area of their AOI exceed the limits of the subregion where they are located. The solution for this situation is to force B to send the updating messages not only to supernode A, but also to supernodes D and E, as discussed below. In the same way, avatar F must send updating messages to the four supernodes.

Following with the same situation, figure 1-right shows the result of the proposed awareness scheme when it is applied to a larger DVE system composed of thirty avatars. It shows the behavior of the proposed technique and how the

different levels of the quad-tree structure are dynamically generated. In this figure, there are several avatars whose AOI intersect with different regions or subregions. These are critical avatars.

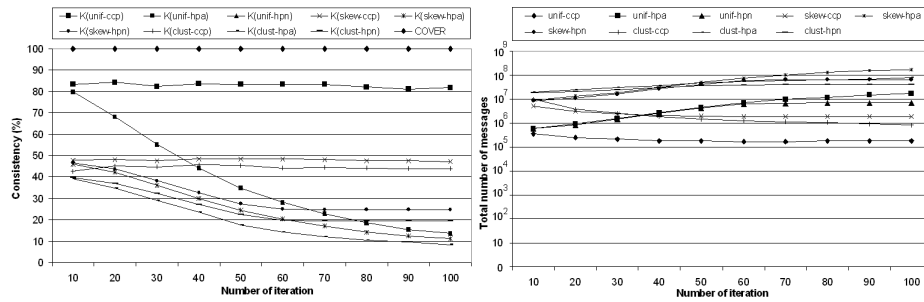
## 4 Performance Evaluation

We propose the evaluation of generic DVE systems by simulation. Concretely, (as we did for the case of DVE systems based on networked-server architectures [25]) we have developed a standalone simulation tool, denoted as SimPeerDve (SPD), that models the behavior of a generic DVE system based on a P2P architecture as a set of independent avatars. These avatars are located within a seamless 3D virtual world [1] following three different and well-known initial distributions: uniform, skewed and clustered [15, 25]. Starting from these initial locations, in each simulation avatars perform 100 iterations. Each iteration consists of each avatar independently moving into the scene. Iterations are performed at the typical rate of 1 avatar movement every 2 seconds [15, 24, 23]). We have considered three different movement patterns: Changing Circular Pattern (CCP) [3], HP-All (HPA) [9] and HP-Near (HPN) [19]. CCP considers that all avatars in the virtual world move randomly around the virtual scene following circular trajectories. HPA considers that there exists certain “hot points” where all avatars approach sooner or later. This movement pattern is typical of multiuser games, where users must get resources (as weapons, energy, vehicles, bonus points, etc.) that are located at certain locations in the virtual world. Finally, HPN also considers these hot-points, but only avatars located within a given radius of the hot-points approach these locations. We have chosen the number of 100 iterations (movements) for a simulation because it is the number of movements that the most distant avatar needs to reach the center of the square virtual world. For evaluation purposes, we have considered the nine possible combinations of the three initial distributions of avatars in the virtual world and the three movement patterns.

Using SimPeerDve, we have performed experimental studies to evaluate the performance of the proposed technique. For comparison purposes, we have simulated the awareness method proposed in the previous section and also the awareness method proposed in [11], since this method currently provides the best awareness results for DVE systems based on P2P architectures (as stated above, the method proposed in [10] has not been evaluated). In order to ensure that the evaluation is performed under the worst case, SimPeerDve allows the overlapping of different avatars at the same location of the virtual environment. Although this situation would be erroneous in a real environment, it allows us to increase the number of avatars located in a given region of the virtual world beyond the limits of a real environment. If awareness is provided under such circumstances, then awareness is guaranteed under real conditions.

Figure 2-left shows the evaluation results for the awareness method proposed in this paper (labeled as COVER) as well as for the awareness method described in [11] (labeled as  $K(x-x)$ ), under all the possible combinations of initial distributions and movement patterns of avatars. This figure shows a representative

example of the experiments performed with a DVE system composed of 1000 avatars. On the X-axis this figure shows the iteration number of the simulation performed, and on the Y-axis it shows the percentage of awareness. This figure shows that for a large scale DVE configuration the method described in [11] only is able to provide a high percentage of awareness (around a 85%) under the uniform movement pattern. For the rest of combinations of initial distributions and movement patterns of avatars, the method proposed in [11] provides a awareness rate below 50%. The worst results of this method are provided for the combination of a clustered initial distribution of avatars and HPA movement pattern, being lower than 10% at the end of the simulation. The reason for this behavior is that for non uniform movement patterns, avatars are unevenly distributed in the virtual world most of the simulation time. Under this situation, the probability that a given avatar  $i$  has one or more unknown neighbors crossing its AOI increases, since its known neighbors of  $i$  also tend to be unevenly distributed around the AOI of  $i$ . The use of supernodes avoids inconsistencies under such situations when using COVER method.



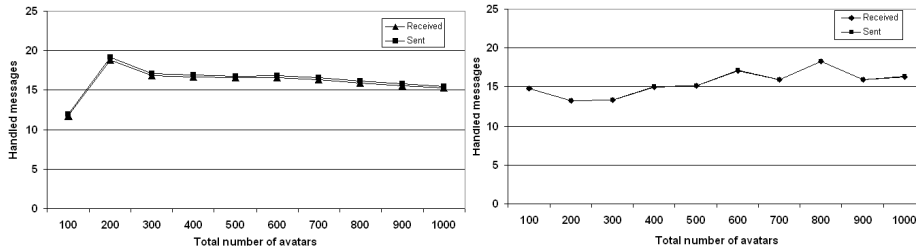
**Fig. 2.** Left: Percentage of awareness provided for all combinations of initial distributions of avatars and movement patterns. Right: Total number of messages exchanged by avatars during the simulations when using COVER method.

In order to show that COVER method provides total awareness without increasing the number of messages exchanged by avatars, figure 2-right shows the number of messages sent during the simulations whose awareness rates are summarized in figure 2-left. The plots in this figure show the number of messages exchanged when using COVER method for the different combinations of initial distributions and movement patterns of avatars. Since the method proposed in [11] limits the number of neighbors that a given avatar can communicate with, the number of messages sent when using this method is not comparable with the number of messages sent when using COVER method. All the simulations whose results are shown in this figure have been performed in a DVE system composed of 1000 avatars. Each point in the plots represents the average value of messages sent by all the avatars in the DVE system during the last 10 iterations.

In order to show the scalability of COVER method, we have also studied the number of messages sent and received by supernodes under different movement



patterns and for simulations performed with different numbers of avatars. For the sake of shortness, we present here the results for those combinations of initial distributions and movement patterns of avatars that show the largest (skewed-HPA) and the smallest (unif-CCP) number of messages exchanged among avatars in figure 2-right. Concretely, figure 3-left shows the average number of messages handled (sent or received) by each supernode in the system during the simulations performed under the combination of a uniform initial distribution of avatars and CCP movement pattern. Each point in these plots is computed as follows: after each iteration in a simulation, the number of supernodes in the system as well as the number of messages sent and received by supernodes are counted, and the *average number of messages per supernode* (ANMS) is computed. Since the number of supernodes dynamically varies, when the simulation finishes, the average value of the 100 ANMS values (each simulation is composed of 100 iterations) is computed. This is the value represented in each point in the plots.



**Fig. 3.** Number of messages handled by supernodes for the uniform-CCP (left) and skewed-HPA (right) combinations

Figure 3-right shows the average number of messages handled by each supernode in the system during the simulations performed under the skewed-HPA combination, the one requiring the largest number of messages exchanged among avatars. The shape of the two plots in this figure is similar to those in figure 3-left, showing a flat slope. The average number of messages handled by each supernode remains within a range of values between 13 and 19, showing that the proposed method scales well with the number of avatars in the system. Moreover, when comparing this figure with figure 3-left, we can see that this range of values is approximately the same for both figures. That is, the behavior of the proposed method does not depend on the movement pattern nor on the initial distribution of avatars in the virtual world. This feature is a key issue to provide an actually scalable awareness method, but it has not been shown in any of the currently proposed methods.

## 5 Conclusions and Future Work

In this paper, we have proposed a new awareness method for large-scale DVE systems based on P2P architectures, denoted as COVER. Unlike the currently

proposed methods, COVER uses two kind of neighbors to provide awareness, normal neighbors and supernodes. While the former ones provide awareness to avatars in the same way other methods do, the latter ones allow to provide awareness to those avatars that are isolated or not completely surrounded by other avatars. This key feature allow to provide full awareness (awareness to 100% of avatars), regardless of the distribution of avatars in the virtual world.

Performance evaluation results show that the proposed method is able to provide full awareness to large scale DVE systems composed of up to 1000 avatars, regardless of both the movement pattern and the initial distribution of avatars in the virtual world. This results have not been shown by any of the currently proposed methods. Due to the quad-tree segmentation algorithm used to select new supernodes, neither the movement pattern of avatars nor the initial distribution of avatars have a significant effect on the number of messages sent by avatars as simulations proceed. Evaluation results also show that this number remains with a flat slope, even for those movement patterns that in the last iterations tend to group avatars in certain points of the virtual world. This result indicates that the proposed method properly balances the workload generated to provide awareness to all avatars. Also, performance evaluation results show that the number of messages handled by supernodes does not increase as new avatars are added to the DVE system. This scalability is achieved by selecting new supernodes when MNUA parameter is exceeded and merging several supernodes into a single supernode when adjacent supernodes manage less than MNUA avatars.

As a result, we can conclude that COVER method provides full awareness to large-scale DVE systems based on P2P architectures in a scalable and efficient way.

## References

1. T. Alexander. *Massively Multiplayer Game Development II*. Charles River Media, 2005.
2. Anarchy Online: : <http://www.anarchy-online.com>.
3. N. Beatrice, S. Antonio, L. Rynson, and L. Frederick. A multiserver architecture for distributed virtual walkthrough. In *ACM VRST'02*, pages 163–170, 2002.
4. C. Bouras, D. Fotakis, and A. Philopoulos. A distributed virtual learning centre in cyberspace. In *Proc. of Int. Conf. on Virtual Systems and Multimedia (VSMM'98)*, November 1998.
5. Everquest: <http://everquest.station.sony.com/>.
6. E. Frecon and M. Stenius. Dive: A scalable network architecture for distributed virtual environments. *Distributed Systems Engineering Journal*, 5(3):91–100, September 1998.
7. R. M. Fujimoto and R.M. Weatherly. Time management in the dod high level architecture. In *Proceedings tenth Workshop on Parallel and Distributed Simulation*, pages 60–67, 1996.
8. L. Gautier and C. Diot. Design and evaluation of mimaze, a multi-player game on the internet. In *Proceedings of IEEE Multimedia Systems Conference*, 1998.

9. C. Greenhalgh, Adrian Bullock, Emmanuel Frion, David Llyod, and Anthony Steed. Making networked virtual environments work. *Presence: Teleoperators and Virtual Environments*, 10(2):142–159, 2001.
10. S. Y. Hu and G. M. Liao. Scalable peer-to-peer networked virtual environment. In *ACM SIGCOMM 2004 workshops on NetGames '04*, pages 129–133, 2004.
11. Y. Kawahara, T. Aoyama, and H. Morikawa. A peer-to-peer message exchange scheme for large scale networked virtual environments. *Telecommunication Systems*, 25(3):353–370, 2004.
12. J. Keller and G. Simon. Solipsis: A massively multi-participant virtual world. In *Proceedings of Parallel and Distributed Processing Techniques and Applications (PDPTA)*, pages 262–268, Las Vegas, USA, 2003.
13. B. Knutsson, H. Lu, W. Xu, and B. Hopkins. Peer-to-peer support for massively multiplayer games. In *Proceedings of IEEE InfoCom'04*, 2004.
14. Lineage: <http://www.lineage.com>.
15. John C.S. Lui and M.F. Chan. An efficient partitioning algorithm for distributed virtual environment systems. *IEEE TPDS*, 13, 2002.
16. John C.S. Lui, M.F. Chan, and K.Y. Oldfield. Dynamic partitioning for a distributed virtual environment. Technical report, Department of Computer Science. Chinese University of Hong Kong, 1998.
17. Michael R. Macedonia. A taxonomy for networked virtual environments. *IEEE Multimedia*, 4(1):48–56, 1997.
18. Michael R. Macedonia, M. Zyda, David R. Pratt, Donald P. Brutzman, and Paul T. Barham. Exploiting reality with multicast groups: A network architecture for large-scale virtual environments. In *Proceedings of the 1995 IEEE Virtual Reality Annual Symposium*, pages 2–10, 1995.
19. M. Matijasevic, K. P. Valavanis, D. Gracanin, and I. Lovrek. Application of a multi-user distributed virtual environment framework to mobile robot teleoperation over the internet. *Machine Intelligence & Robotic Control*, 1(1):11–26, 1999.
20. D.C. Miller and J.A. Thorpe. Simnet: The advent of simulator networking. *IEEE TPDS*, 13, 2002.
21. D. Milojevic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-peer computing. Technical report, Technical Report HPL-2002-57, HP Laboratories, Palo Alto, 2002.
22. S. Mooney and B. Games. *Battlezone: Official Strategy Guide*. BradyGame Publisher, 1998.
23. P. Morillo, J. M. Orduña, M. Fernández, and J. Duato. An adaptive load balancing technique for distributed virtual environment systems. In *Proc. of Intl. Conf. on Parallel and Distributed Computing and Systems (PDCS'03)*, pages 256–261. IASTED, ACTA Press, 2003.
24. P. Morillo, J. M. Orduña, M. Fernández, and J. Duato. A fine-grain method for solving the partitioning problem in distributed virtual environment systems. In *Proc. of Intl. Conf. on Parallel and Distributed Computing and Systems (PDCS'04)*, pages 292–297. IASTED, ACTA Press, 2004. Best paper award in the area of load balancing.
25. P. Morillo, J. M. Orduña, M. Fernández, and J. Duato. Improving the performance of distributed virtual environment systems. *IEEE Transactions on Parallel and Distributed Systems*, 16(7):637–649, 2005.
26. M. Oliveira, J. Crowcroft, and M. Slater. Components for distributed virtual environments. *PRESENCE, The MIT Press*, 10(1):56–61, 2001.
27. Quake: <http://www.idsoftware.com/games/quake>.

28. D. Roberts and R. Wolff. Controlling consistency within collaborative virtual environments. In *Proceedings of IEEE Symposium on Distributed Simulation and Real-Time Applications (DSRT'04)*, pages 46–52, 2004.
29. J.M. Salles, Ricardo Galli, and A. C. Almeida et al. mworld: A multiuser 3d virtual environment. *IEEE Computer Graphics*, 17(2), 1997.
30. S. Singhal and M. Zyda. *Networked Virtual Environments*. ACM Press, 1999.
31. J. Smed, T. Kaukoranta, and H. Hakonen. A review on networking and multiplayer computer games. Technical report, Turku Centre for Computer Science. Tech Report 454., 2002.
32. Randall B. Smith, R. Hixon, and B. Horan. *Collaborative Virtual Environments*, chapter Supporting Flexible Roles in a Shared Space. Springer-Verlag, 2001.
33. Starcraft: <http://www.blizzard.com/starcraft>.
34. A. Steed and C. Angus. Supporting scalable peer to peer virtual environments using frontier sets. In *IEEE Virtual Reality-2005*. IEEE Computer Society, 2005.
35. P.T. Tam. Communication cost optimization and analysis in distributed virtual environment. Technical report, Department of Computer Science. Chinese University of Hong Kong, 1998.
36. S. Zhou, W. Cai, B. Lee, and S. J. Turner. Time-space consistency in large-scale distributed virtual environments. *ACM Transactions on Modeling and Computer Simulation*, 14(1):31–47, 2004.