

MOPAR: A Mobile Peer-to-Peer Overlay Architecture for Interest Management of Massively Multiplayer Online Games

Anthony (Peiqun) Yu
University of British Columbia
2366 Main Mall
Vancouver, BC, V6T 1Z4, Canada
1-604-822-2145
pqyu@cs.ubc.ca

Son T. Vuong
University of British Columbia
2366 Main Mall
Vancouver, BC, V6T 1Z4, Canada
1-604-822-6366
vuong@cs.ubc.ca

ABSTRACT

In this paper we propose a fully distributed peer-to-peer (P2P) infrastructure supporting Networked Virtual Environment (NVE) applications, such as massively multiplayer online games (MMOG). While many attempts have been made to tackle one of the most challenging issues in MMOGs - interest management, none of them are considered truly successful. Our architecture is a hybrid scheme focusing on NVEs' interest management. Our scheme takes the advantage of both structured overlay, i.e. Distributed Hash Table (DHT), and the unstructured P2P architecture. It not only has more stable and consistent performance with respect to neighbor discovery, but also is more scalable and fault tolerant than the existing approaches. Unlike other hexagonal zoning approaches in which each participant has a discrete view of the virtual world, our zoning design guarantees that all participants have a continuous view. Moreover, our novel hierarchical architecture and the message dissemination algorithm greatly save network bandwidth and alleviate each node's workload. We implemented our infrastructure running on top of an emulated network. We also implemented a simple game simulation and a visualization tool to demonstrate and visualize our infrastructure.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design – Distributed networks

General Terms

Algorithms, Performance, Design

Keywords

Peer-to-Peer, Networked games, Networked Virtual Environment, interest management, scalable, massively multiplayer online games.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NOSSDAV'05, June 13–14, 2005, Stevenson, Washington, USA.
Copyright 2005 ACM 1-58113-987-X/05/0006...\$5.00.

1. INTRODUCTION

Massively multiplayer online games (MMOG) are becoming popular nowadays. They share many common characteristics and issues with Networked Virtual Environments (NVE) [15]. Among all these common issues, scalability is the most concern in the commonly used client-server model today. In typical MMOGs, for example, EverQuest [5], hundreds and thousands of participants play their roles in the virtual world at the same time. However, in the traditional client-server model, scalability can be problematic. One of the obvious reasons is that it's very hard for a game operator to predict the number of users when starting a game service. In addition, even in the same day, demands for the service can also frustrate. Therefore, there is no easy way to determine the right investment on the server beforehand.

Recently, peer-to-peer architectures have been proposed to tackle the scalability issue of the MMOGs [3, 9, 12, 13, 17, 18]. P2P's properties such as resources sharing and decentralized resource consumption are considered to be very useful for addressing the scalability issue. Every node joining the system can bring more resources to the virtual environments, so that with more users the game has more resources.

However, while P2P remedy can address some issues, it also brings some side effects. In the NVEs, each participant has limited visibility, for an instance, the area within 100 meters of distance. Therefore, broadcasting the messages to all the participants in the virtual world is definitely not bandwidth efficient, and wastes the computing power of those nodes that receive irrelevant messages. Typically each participant is only interested in the activities happening within its Area of Interest (AOI). Interest management [10] is commonly used in the large scaled virtual environment. By doing so, network bandwidth consumption is greatly reduced, and the workload of each participant is also alleviated. In the traditional NVEs, interest management is done either through server-side filtering, or using the IP multicast where a multicast group represents an area of the NVE [12]. While interest management is straightforward in the client-server model through message filtering, it is extremely challenging in the pure P2P system.

In this paper, we propose an infrastructure supporting P2P interest management. In our approach, each node in the virtual world dynamically builds connections with its neighbors; those participants appear in the node's AOI. This is called neighbor discovery. Each participant then directly communicates with its neighbors without a coordinator. The main contribution of this

paper is that we propose a pure P2P interest management scheme, which is scalable and has high capability of fault tolerance. The infrastructure also adds low overhead over each node for the neighbor discovery.

The paper is organized as follows: After discussing the related work in Section 2, we present our overall design in Section 3. We then discuss our neighborhood dissemination algorithm in Section 4. Our implementation and visualization tool are described in Section 5. At the end, we provide an analysis of our proposal in section 6, and then conclude the paper in section 7.

2. RELATED WORK

2.1 DHT vs. Unstructured P2P Approaches

In recent years, many proposals have been made to improve the NVEs' scalability using P2P architectures. They can be categorized into two kinds of approaches. One uses structured P2P such as a distributed hash table (DHT), while the other kind is built on top of an unstructured P2P architecture similar to Gnutella.

In the DHT approach, such as Knutson et al's P2P Support for Massively Multiplayer Games [3] and Limura et al's Zoned Federation of Games Servers [17], they divide the game map into fixed regions. An authoritative server is dynamically assigned for each region responsible for coordinating the game states. This kind of approach aims to emulate the cluster of servers by dynamically promoting peer nodes to be coordinators. While Knutson et al proposes to use Scribe [11], an application level multicast built on top of Pastry [2], for the game state dissemination, Limura et al argues that Scribe incurs unnecessary network delay due to the possible number of hops. They propose a direct communication model, in which coordinators cache the game states and maintain direct connections with region members. Coordinators then can be used for storing members' states and updating members' states through the direct communication between them and region members. In Limura et al's Zoned Federation of Games Servers, coordinators are selected through a registration process via the DHT. We use the similar approach for the election of the master node for a cell, however, our master node election process is much simpler, and the functionalities of a master node are entirely different from that of a coordinator.

The common problem of this kind of approach is that they do not address the AOI of a player that may be outside of a region. Each participant is put in an artificially partitioned region as depicted in Figure 1. Participants in different region cannot view each other even they are within each other's AOI in the virtual world.

Another problem is that it's hard to determine the size of each region and the optimal way of partitioning beforehand. Not only the game applications are very different among each other, but also the game states in a single game can be fairly dynamic. Crowds of participants can appear in one area at one time, then in another area a short time later. Moreover, because game states communications must be relayed through coordinators, the coordinators can become a bottleneck. If there are too many players appear in one region, the region coordinators can be easily overloaded.

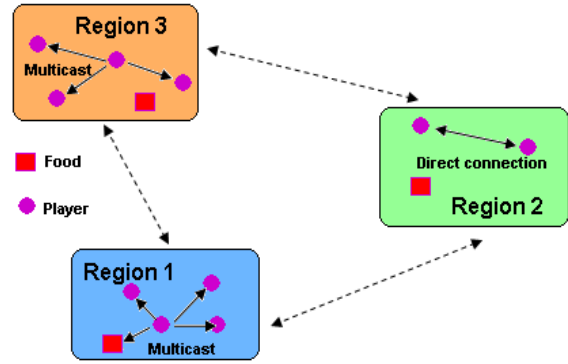


Figure 1. Map is partitioned into fixed regions [3]

The second kind of approach is based on unstructured P2P structure, in which all the nodes in the NVEs must connect their neighbors even they are dispersed in the virtual world. In this type of scheme, the challenge is to guarantee both global connectivity and local connectivity in an efficient way [9].

There are three majors proposals fall that in this category. They are Kawahara et al's A Peer-to-Peer Message Exchange Scheme [18], Keller, et al's Solipsis [9], and Hu et al's Voronoi scheme [13]. One of the common problems of these proposals is the inefficient use of the resources. In Kawahara et al's approach, each participant continuously exchange neighbor list with its nearest neighbors. The messages generated for neighborhood maintenance grow at $O(n^2)$, where n is the average number of neighbors of each participant [13]. In Solipsis [9], every participant acts as a "watchman" of others. Each participant watches every movement of their neighbors, and notifies others when new neighbors are discovered for them. The main difference between Exchange Message Scheme and Solipsis is that, in Solipsis, each participant connects to the neighbors who may not be nearest in the game world. In Voronoi scheme [13], every participant needs to look ahead through the boundary neighbors who may be outside of its AOI to maintain the consistent topology for every single movement. One of the major problems of the above proposals is that movement is the most common activities in a game. It's not efficient to make every game participant involve in the interest management and be a "watchman" for others.

The most significant problem of these schemes is that, none of them are successful on maintaining the consistent topology, or discovering neighbors. In Message Exchange Scheme [18], participants exchange neighbor list only with their nearest neighbors. As a result, a participant or a group of participants can be isolated, if they are far away from other participants in the game world [9]. Solipsis [9] can not guarantee the neighbor discovery either, since the directly connected neighbors still have chances of missing incoming nodes in some cases [13]. Voronoi scheme [13] can hardly maintain the consistent topology, in case two or more adjacent boundary neighbors leave the game at the same time.

2.2 Hexagonal Zoning

Hexagonal zoning is widely used in the game applications. For example, in Macedonia et al's proposal [12], a participant's AOI consists of a radius of grid cells, where a participant joins new cells

at the leading edge and leaves old cells at the trailing edge as it moves forward. Each cell is mapped to a multicast group.

Hexagonal zoning has many advantages. Hexagons have uniform orientation and uniform adjacency. A participant moving from one cell to another joins and leaves same number of cells (Figure 2). Because an AOI is usually defined by radius, hexagonal zoning is also more approximated to circle than quadrant zoning.

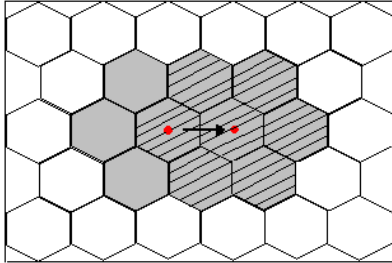


Figure 2. The gray area represents the AOI of a participant. A participant moving from one cell to another joins and leaves the same number of cells.

However, in the traditional hexagonal zoning, the view of each participant in the virtual world is discrete; the view will not change until it moves from one cell to another cell. The discrete view seems unrealistic in the NVEs, especially for these applications in which, the participants' visibility tends to be fairly sensitive.

3. A HYBRID OF DHT AND UNSTRUCTURED P2P

In this section, we describe our novel interest management scheme. This scheme takes advantage of both DHT and unstructured P2P structure, so it can be considered as a hybrid of the two approaches discussed above.

3.1 Hexagonal Zoning with Continuous View

Although we also use hexagons to divide our game map, our design is different from Macedonia's proposal [12] in three ways:

- First, we don't rely on a special network support such as IP multicast. We use DHT overlay to map each cell to a determined node. We will elaborate it in Section 3.1.
- Secondly, when a game participant is moving from one cell to another, it does not have to join or leave other cells, unless it is the master node for the current cell and/or will become the master node for the entering cell. It will be clear after the following sections.
- The view of each participant in the virtual world is continuous. We will describe this in Section 3.2.

In order to simplify the scenarios, we assume that each participant has the same and fixed size of AOI.

3.2 Building Hierarchical Structure Using a DHT

There are many variations of DHT such as Pastry [2], CAN [14], Chord [7], and Tapestry [4]. We choose Pastry for our scheme since

its use of a leaf set naturally supports the dynamic characteristic of NVEs. The nodes in the Pastry leaf set are the numerically closest nodes to the current node and can be reached from a neighboring node in one hop (directly). This property is very useful for the state replication, and also for a newly joined node to immediately locate and obtain the states belonging to it. This will be elaborated in later sections.

3.2.1 The Role of Master, Slave and Home Nodes

We introduce three types of nodes based on their roles in our interest management scheme. They are master nodes, slave nodes and home nodes. Master nodes and slave nodes have a position in the same cell. Each cell has at most one master node, but can have multiple slave nodes. Unlike master and slave nodes, home nodes are virtual nodes, which means they are not necessarily positioned in the same cell as master or slave nodes. Because of the random mapping via SHA-1, it is unlikely that a master or a slave node of a cell is also the home node for the cell. If there is no joining or leaving node, the correspondence between a cell and its home node is bounded, however, master and slave nodes can change their belonging cells while they are moving in the virtual world.

When a participant first joins into the NVE, following procedures are preceded:

1. Map the participant's coordinates to the Cell ID [6]. For example, a participant at position (30, 50) can be mapped to the hexagon cell ID (0, 0), if the side length of the hexagon equals to 60.
2. Hash the Cell ID to the 128-bit ID by applying SHA1-1 hash function [3]. We call it HexId.
3. Query the Pastry overlay to obtain the node N whose Pastry Id is numerically closest to the HexId. Node N is now identified as the home node for this cell.
4. The home node is used for master node registration. Each cell can have at most one master node. If a newly joined node finds that there is no master node registering for this cell after querying the home node, it registers itself as the master node; otherwise, it becomes the slave node.
5. After the above processes, the newly joined node then either waits for slave nodes to subscribe to it, or subscribes to a master node, depending on whether it's a master node or slave node.
6. The master node needs to query the home nodes of the neighboring cells for building the direct connections with the neighboring master nodes.

At this stage, our hierarchical structure has been built. In this structure, master nodes play the main role for the interest management. They communicate with the master nodes in the neighboring cells, and notify the slave nodes their neighbors within their AOI.

3.2.2 Handling the Dynamic Characteristics of NVE

Unlike typical applications for DHTs such as file sharing [1], NVEs are much more dynamic in terms of frequency of the node joining and leaving the environment.

When a node first joins the environment, it can become the home node for some cells immediately. This situation can happen when the newly joined node's Pastry ID becomes numerically closest to those cells' HexIds. For example, $[N0-C] < [N1-C]$, where $N0$ is the newly joined node's Pastry ID, C is a hashed cell ID, and $N1$ is the original home node for the cell C . We observe that, if a newly joined node becomes a home node for some cells, the original home nodes for these cells must be the closest nodes on either or both side of $N0$. Fortunately, the Pastry's leaf set maintains the information about nearest left node and right node. Therefore, a newly joined node sends a message to these two nodes for retrieving the qualified master nodes registration records. On receiving the request, the nearest nodes scan their storage to check if they should pass the ownership of master nodes for some cells. If yes, they transfer these master node registration records over to the requesting node. Figure 3 describes the relationship between the newly joined node and the nearest nodes in the leaf set.



Figure 3. Node $N0$ represents the newly joined node. $N1$ and $N2$ are the nearest nodes on $N0$'s left and right side.

The registration records of master nodes are always replicated to the k next most qualified home nodes. That is the node whose ID is the next closest to the current home node's ID (to the left and right in Pastry IDs – this is typically done in file storage [1]). Whenever a node leaves the environment a replica node will immediately become the home node for the master node. The replication factor k can be increased to cope with the high failure network; however, the invariant for the number of replicas should always be kept.

3.3 Unstructured P2P Communication Scheme

Although we use DHT overlay as the support for building the hierarchical structure, in order to minimize the overhead incurred by the DHT overlay, we use an unstructured P2P scheme for maintaining the neighborhood topology and discovering neighbors in a hierarchical way.

In other unstructured P2P approaches we discussed earlier, participants maintain the neighborhood topology and discover neighbors by either periodically exchanging neighborhood information with the current neighbors, or watching the new neighbors for each other in a flat model. We apply the same idea in the hierarchical fashion. In our scheme, only master nodes exchange neighbor list with other master nodes in the neighboring cells, and watch the new neighbors for the slave nodes. Slave nodes are notified only if their neighbor sets have any changes, i.e. there are some new neighbors moving into the AOI. In order to further reduce the messages between the master nodes and slave nodes for the movement updates, our master nodes have the ability to predict the positions of the slave nodes. A slave node only needs to update the master node when its moving function (direction or speed) has changed.

A participant who first joins the game gets the neighbor set through one aggregated message from the master node immediately, reducing the delay for the joining process comparing to the flat model.

Compared to the regular hexagonal zoning, in our scheme, the view of each participant in the virtual world is continuous instead of discrete. That is because the AOI of each participant is smaller than the covering area of the master nodes. As a specific example shown in Figure 4, a master node's viewing area covers 7 cells, while a slave node's viewing area is within the 7 cells, therefore, a master node can provide continuous neighborhood information to every participant in its governing cell. A master node's viewing area is extensible to cover the slave nodes' AOI.

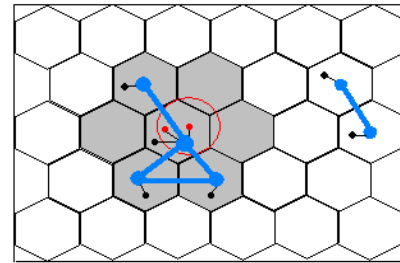


Figure 4. The gray area represents the covering area of the master node in the center cell of this area. Lines between nodes represent the communications for interest management. The circle represents the AOI of the slave node in the center of the circle.

As the Figure 4 shows, our communication scheme is similar to the unstructured P2P interest management scheme, but slave nodes only need to communicate with the master nodes. Note that we split the communications into two types. One is related to the interest management as we discuss here; the other is the game states communication which is outside of our AOI system. Participants still need to communicate directly with others within their AOI for the game interaction.

3.4 Minimizing the Disadvantages of Both DHT and Unstructured P2P Approaches

Since a DHT has overhead for the $O(\log(n))$ hops from the source node to the destination node, our design goal is to minimize the use of the DHT for the interest management communications. In our scheme, the DHT is used only for maintaining the hierarchical structure, or registering master nodes. If the master nodes rarely leave a cell, and every cell has a master node, the DHT will not be used. Slave nodes can obtain the direct connections with the master nodes of the neighboring cells through the current master node, when they want to move to the neighboring cells. Although it is an ideal case, participants in MMOG tend to move slowly and have a tendency of gathering together.

On the other hand, it's very challenging for the pure unstructured P2P architecture to maintain global connectivity. Global connectivity makes sure that any participant or a group of participants can never be isolated [9]. In fact, none of them are can successfully maintain the global connectivity as discussed earlier. However, in our scheme, global connectivity can never fail,

because a node can always connect to another node, through the assistance of home nodes if necessary.

4. NEIGHBORHOOD DISSEMINATION ALGORITHM

The connections between master nodes are the backbone for neighborhood dissemination. Slave nodes subscribe to corresponding master nodes for the notifications of any neighborhood changes. A new master node gets in touch with the other master nodes in the neighboring cells via DHT lookup, and builds direct connections with them. When a master node leaves a cell, it will assign the most capable slave node in the cell as the next master node, and pass all the relevant information over to the newly assigned master node.

Slave nodes only need to update the corresponding master nodes, when their moving functions (speed or direction) are changed. Compared to the traditional constant position updates, our algorithm greatly reduces the number of updates, based on our assumption that a player's moving speed and direction should not change constantly. Master nodes then predict each participant's current position based on the participant's moving function [16]. Neighboring master nodes periodically exchange the participant list and these participants' positions among each other.

Master nodes do not need to update slave nodes, unless new neighbors move into the slave nodes' AOI. Slave nodes communicate with the neighbors within their AOI continuously for the game interaction. Therefore, they do not need the master nodes' involvement for notifying the accurate positions of the neighboring participants. In the same manner, they also know when a participant is leaving their AOI.

This neighborhood dissemination algorithm aims to further minimize the message cost for the interest management in addition to the hierarchical structure, which is successful, since the algorithm fully utilizes the moving function and game states interaction to minimize the interest management messages.

5. IMPLEMENTATION AND VISUALIZATION

We implemented this infrastructure on top of FreePastry [2], which is an open source version of Pastry. We use the network emulator provided by the FreePastry as the network environment. All the nodes are running on the same Java VM. The capacity for the number of concurrent participants is constrained by the memory available. Our infrastructure, MOPAR is another layer added on top of FreePastry.

In order to show the effect of the neighbor discovery, we also implemented a simple game simulation and a visualization tool (Figure 5). The game is simple; participants move randomly on the map. There is one activated player at a time. The activated participant sends message to others who fall in its AOI. The participants who receive the message blink by changing color between black and red constantly. In the visualization tool, the master nodes distinguish themselves by doubling their size relative to slave nodes. One can easily view how a node switch role between slave and master node dynamically. The hexagon zoning can be shown as the background of the map.

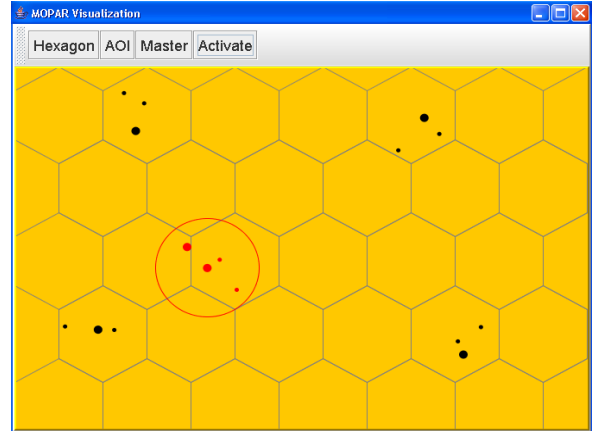


Figure 5. Game simulation and infrastructure visualization. The bigger dots represent the master nodes, while the smaller dots represent the slave nodes. The circle indicates the AOI of a participant.

6. ANALYSIS

MOPAR tends to be more scalable than other schemes due to the hierarchical structure and the efficient neighborhood dissemination algorithm. We have fewer participants, only the master nodes exchange the neighbor list compare to the Message Exchange Scheme [18]. We also have fewer "watchmen" compared to Solipsis [9], and Voronoi scheme [13].

The resources in our scheme are used in an efficient way. In the unstructured P2P schemes, participants must connect to their neighbors; even they are dispersed on the map. This is necessary for their schemes, because otherwise global connectivity cannot be maintained. However, it's common that there are clusters of participants in the NVEs. Participants who are doing different activities in different virtual locations do not need to be aware of each other. Our scheme allows them be entirely separated without worrying about any node being isolated.

Our scheme is also more fault tolerant than the pure unstructured P2P schemes through the use of the DHT. Any fault caused by the failure of one node can be resolved through DHT structure. Other schemes might need a small server to support fault tolerance. Our system is purely peer-to-peer.

Although there is some overhead for the master nodes to register via DHT, this is minimal. Typically, in an NVE such as MMOG, participants are gathering together for some activities, the ratio of master nodes by slave nodes should be fairly low. Most importantly, our scheme guarantees both of the global and local connectivity in an efficient manner.

Moreover, it's hard for the pure unstructured P2P schemes to address game states persistency and consistency issues. Although addressing these issues is left for future work, they can be addressed easily, through the use of a DHT.

7. CONCLUSIONS AND FUTURE WORK

We proposed a novel hybrid infrastructure supporting interest management, one of the most challenging problems in peer-to-peer networked virtual environments. Our scheme takes advantage of

both the DHT overlay and the unstructured P2P architecture. We divided the game map into hexagonal zones. Each zone has a corresponding home node via the DHT mapping. Each cell has at most one master node. The master node registers itself to the home node of the cell. The connections between the master nodes act as the backbone for the neighborhood dissemination. Slave nodes in each cell receive the neighborhood update from the master nodes. Our design supports a continuous view for nodes in the virtual world. The hierarchical structure and the efficient neighborhood message dissemination algorithm greatly save the bandwidth of the network. Our system is scalable, fault tolerant, and bandwidth efficient.

We have also implemented a simple game simulation and a visualization tool, in which, participants move randomly and communicate with neighbors. The visualization tool allows users to interact with the infrastructure to explore and understand the design of the hexagonal zoning, and the roles of master nodes and slave nodes in the infrastructure.

Our future work will improve the load balancing, and the granularity of the hexagonal zoning. When a master node does not have many slave nodes in its cell, the merging of master nodes for adjacent cells should be considered [8]. If there are too many slave nodes appear in one cell, zones may be split to assign more master nodes for a given area to increase scalability. In addition, in order further reduce the DHT lookups; we will build direct connections among the adjacent home nodes, so that a master node's registration can be disseminated through the directly connected home nodes. By doing so, a master node can obtain the neighboring master nodes information from its home node directly without too many DHT lookups. The role switching between master and slave, for example, when a participant moves from one cell to another, may also be optimized by eliminating some unnecessary switching. Moreover, we will also consider extending our infrastructure to address states persistency and consistency issues.

8. ACKNOWLEDGEMENTS

We would like to thank Dr. Charles Krasic, Mike Blackstock, and Kamran Malik for their useful comments and valuable feedbacks, and give special thanks to Mike Blackstock for proofreading.

9. REFERENCES

- [1] A. Rowstron, P. Druschel. Storage management and caching in Past, a large-scale persistent peer-to-peer storage utility. In Proc. ACM SOSP'01, Banff, Canada, Oct. 2001.
- [2] A. Rowstron et. al. Pastry: Scalable decentralized object location, and routing for large-scale peer-to-peer systems. In Proceedings of the 18th IFIP/ACM international conference on Distributed Systems Platforms (Middleware). November 2001.
- [3] B. Knutsson, H. Lu, W. Xu, B. Hopkins. Peer-to-peer Support for Massively Multiplayer Games. In INFOCOM, Mar, 2004.
- [4] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-0101141, UC Berkeley, April 2001.
- [5] EverQuest: <http://everquest.station.sony.com/>
- [6] GameDev.net: Coordinates in Hexagon-Based Tile Maps <http://www.gamedev.net/reference/articles/article1800.asp>
- [7] Ion Stoica, Robert Morris, David Karger, Fans Kaashoek, and Hari Balakrishnan, Chord: A scalable Peer-to-Peer lookup service for internet applications. In Roch Guerin, editor, Proceedings of SIGCOMM-01, volume 31, 4 of Computer Communication Review, Pages 149-160, New York, August 27-31 2001. ACM Press.
- [8] J. E. Jaramillo, L. Escobar, and H. Trefftz. Area of Interest Management by Grid-Based Discrete Aura Approximations for Distributed Virtual Environments. In proceedings of the 6th Symposium on Virtual Reality (SVR 2003), pp 342-353, Ribeirao Preto, SP - Brazil, October 15-18, 2003
- [9] J. Keller, G. Simon. Solipsis: A Massively multi-Participant Virtual World. In Proc. of PDPTA 2003. Pg. 262-268. 2003
- [10] Katherine L. Morse. Interest management in large-scale distributed simulations. Technical Report ICS-TR-96-27, University of California, Irvine, 1996.
- [11] Miguel Casro, Michael B Jones, Anne-Marie Kermarrec, Antony Rowstron, Marvin Theimer, Helen Wang, and Alec Wolman, An evaluation of scalabe application-level multicast built using peer-to-peer overlays. In Infocom'03, April 2003.
- [12] M. R. Macedonia, M. J. Zyda, D. R. Pratt, D. P. Brutzman, P. T. Barham. Exploiting Reality with Multicast Groups. IEEE Computer Graphics and Applications, Volume 15, Issue 5, Pages: 38-45, Sept. 1995.
- [13] S. Hu, G. Liao. Scalable Peer-to-Peer Networked Virtual Environment. In SIGCOMM'04 workshops.
- [14] Sylvia Ratnasamy, Paul Francis, mark Handley, Richard Karp, and Scott Schenker, A scalable content-addressable network. In Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, pages 161-172. ACM Press, 2001.
- [15] Shinghal, S., and Zyda, M., Networked Virtual Environments, ACM Press, 1999.
- [16] S. Saltenis, C. S. Jensen, S. T. Leutenegger, and M.A. Lopez. Indexing the positions of continuously moving objects. In Proc. of the 2000 ACM SIGMOD int'l. conf. On Management of Data, pages 331-342, 2000
- [17] T. Limura, H. Hazeyama, Y. Kadobayashi. Zoned Federation of Game Servers: a Peer-to-Peer Approach to Scalable Multi-player Online Games. In SIGCOMM'04.
- [18] Y. Kawahara, T. Aoyama, H. Morikawa. A Peer-to-Peer Message Exchange Scheme for Large-Scale Networked Virtual Environments. Telecommunication Systems Vol. 25 Issue 3, Pages 353 370, 2004