Voronoi-based Adaptive Scalable Transfer revisited

Gain and Loss of a Voronoi-based Peer-to-Peer Approach for MMOG

Cand.-Inform. Helge Backhaus Institute of Telematics
University of Karlsruhe (TH)
Karlsruhe, Germany
helge.backhaus@stud.uni-karlsruhe.de

Dipl.-Inform. Stefan Krause Institute of Telematics
University of Karlsruhe (TH)
Karlsruhe, Germany
stkrause@tm.uni-karlsruhe.de

ABSTRACT

We present and evaluate an implementation of VAST (Voronoi-based Adaptive Scalable Transfer) as proposed by Shun-Yun Hu and Guan-Ming Liao in *Scalable Peer-to-Peer Networked Virtual Environment* [4]. VAST is a fully-distributed peer-to-peer protocol, designed to handle event messages in MMOGs (Massively Multiplayer Online Games) and virtual worlds. VAST relies on voronoi diagrams and AOI (Areas Of Interest) for neighbor discovery. Benefits and problems of using voronoi diagrams in peer-to-peer-based MMOGs and virtual worlds are discussed, by looking at bandwidth usage, scalability and consistency of VAST and comparing it to other approaches. We point out certain issues regarding VASTs consistency and propose some changes to VASTs architecture aiming at enhancing its consistency. Finally we have a look at future extensions like group management and adaptive areas of interest per peer, in order to reduce bandwidth requirements and further increase VASTs consistency.

Categories and Subject Descriptors

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

General Terms

design experimentation performance

Keywords

peer-to-peer, voronoi diagram, scalability, neighbor discovery, massively multiplayer online games, virtual worlds

1. INTRODUCTION

In recent times, MMOGs (Massively Multiplayer Online Games) enjoy great popularity. Additionally, virtual worlds like Second Life are gaining more and more interest even by mainstream media. Most of these games are build upon a centralized client/server architecture. This way, the game provider keeps maximum control over the players. Unfortunately, the centralized structure leads to high maintenance costs and a poor scaleability limiting the number of concurrent players. As a more scaleable solution, peer-to-peer networks were suggested.

In *Scalable Peer-to-Peer Networked Virtual Environment* [4], Shun-Yun Hu and Guan-Ming Liao proposed a voronoi based peer-to-peer network to handle event messages in MMOGs and virtual worlds. To test the viability of this protocol, we implemented it for the overlay simulator OverSim [2] and evaluated its performance and stability under different scenarios.

This paper is organized as follows: In section 2 we present a number of peer-to-peer protocols for MMOGs. In section 3, we take a closer look at the protocol VAST and its basic functionality. In section 4 we evaluate and present the results achieved through different simulations conducted with VAST. In section 5 we point out flaws in VASTs design, which lead to overlay inconsistencies. In section 6 our results are summarized. Finally in section 7 we present different changes and additions to VASTs design, which we want to implement in the future.

NetGames'07, September 19-20, 2007, Melbourne, Australia

2. RELATED WORK

2.1 Application Layer Multicast

Knutsson et. al. proposed and implemented SimMud [6], a simple massively multiplayer game based on a peer-to-peer overlay. SimMud uses DHT (Distributed Hash Tables) by means of Pastry [8] and ALM (Application Layer Multicast) by means of Scribe [9] for event distribution. SimMuds game world is divided into fixed sized regions and SimMud employs so-called coordinators, which manage these regions. A coordinator is the root of a multicast tree and all players in a region subscribe to the corresponding coordinators multicast address. By creating several replicas of the coordinators, fault tolerance is achieved and global states can be saved. Zone changing is achieved by keeping connections between the coordinators, so that players can change from one zone to another through their coordinator.

It can be necessary for a player to subscribe to multiple coordinators in order to communicate with players in other regions. This introduces unnecessary network overhead, because messages from all players of these other regions are received, though a player might just be interested in messages from a few players outside of his region. Due to the nature of DHT and ALM, messages between players are of-ten relayed by other players and therefore SimMud shows quite bad results regarding end-to-end delay.

2.2 Distributed event delivery for MMOG

Shinya Yamamoto et. al. proposed *A Distributed Event Delivery Method with Load Balancing for MMORPG* [12], where the game space is divided in multiple subspaces of fixed size. Each subspace has a responsible player, which is in charge of delivering game events occurring in their responsible subspace. The approach includes a load balancing mechanism. Players responsible for a crowded subspace can construct node trees, which are used for message forwarding and reduce event forwarding overhead per player. Also end-to-end event delay reduction is achieved, by dynamically replacing nodes within the tree. As players can be part of more then one subspace, a subscription system ensures seamless movement and node assignment from one subspace to another.

Even with load balancing and end-to-end event delay reduction mechanisms, unnecessary messages are received and network overhead is high at times, because players subscribe to all subspaces within a specific distance without taking their true AOI (Area Of Interest) into account.

2.3 Zoning Layer

Takuji Iimura et. al. in *Zoned Federation of Game Servers* [5] propose a zoned federation layer, which is inserted between the application layer and the overlay layer. The basic idea behind the zoned federations concept is to divide the game world into multiple game zones, in order to distribute the workload of one main server on different players, called authoritative nodes, each one just responsible for one part of the whole game world. DHTs are used to store global data and states, which are also cached by authoritative nodes. The DHTs enable players to find their authoritative node, when changing zones or for making initial contact upon joining the game. Afterwards the player keeps a connection to his authoritative node, granting him the ability to communicate with all players within his zone through two overlay hops. Global data can only be modified by authoritative nodes. Whenever a player wants to change some global data, it has to do so by sending these changes to his zones authoritative node. Authoritative nodes aggregate modifications from and send state change modifications to their zones members. Provided that some reputation mechanism is present, this grants limited anti-cheating capabilities, by letting only trusted players become authoritative nodes and granting authoritative nodes the option to reject state changes from untrusted players.

End-to-end delay in this approach is very good, but the workload on authoritative nodes can get rather heavy, up to the point where it might be to much to be handled by a game players PC.

3. VORONOI-BASED ADAPTIVE SCALEABLE TRANSFER

3.1 Design

In this section we explain the layout and functionality of VAST as proposed by Shun-Yun Hu and Guan-Ming Liao in *Scalable Peer-to-Peer Networked Virtual Environment.* VAST is based on the assumption that each player can just perceive a very limited portion of the whole game world and thus is only interested in the events in his immediate surroundings called AOI (Area Of Interest). VAST assumes all players to have a circular AOI of exactly the same size. However in [4] the authors already proposed to use adapting AOIs per player of varying shape in the future.

VAST utilizes voronoi diagrams [1] to define three different types of neighbors for a given player. Figure 1 shows these three different types of neighbors VAST distinguishes between.

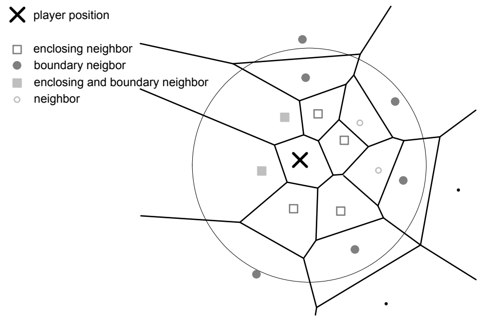

Figure 1: VAST distinguishes between enclosing, boundary and *normal* neighbors.

The voronoi diagram divides the whole game world into arbitrary sized regions based on the players spatial coordinates and therewith assigns to each player his own region. Enclosing neighbors are those whose region shares a common edge with the players own region. Boundary neighbors are defined as regions, which overlap with the players AOI-boundary. Therefore it is possible that an enclosing neighbor is also a boundary neighbor. *Normal* neighbors are those who are neither enclosing nor boundary neighbors.

3.2 Neighborhood discovery

When joining the overlay, each player receives all his current neighbors at his joining position. VASTs basic idea then is to let each node construct his own local voronoi diagram, based on the coordinates of his neighbors and to keep direct connections to all of them.

Neighbor discovery in VAST is mainly handled by a players boundary neighbors. Moving players send position updates to their neighbors. Upon incoming updates boundary neighbors send queries to their enclosing neighbors and check whether the moving players AOI overlaps with their enclosing neighbors regions. New found neighbors are then forwarded to the moving player, who directly connects to them and stores them in a locally managed neighbors list (Figure 2a). Also the moving player disconnects from all boundary neighbors whose regions no longer overlap with his AOI after his movement as shown in Figure 2b).

As players move around the game world, their voronoi diagrams layout is constantly changing and needs to be rebuild by each player on incoming position updates from his neighbors. We use a modified version of Fortune's sweepline algorithm [3] to construct each players voronoi diagram. The algorithm is capable of constructing voronoi diagrams in O(n log n) time, while at the same time calculating neighborhood relationships. Fast construction of each players voronoi diagram is vital to allow low latency, when processing messages

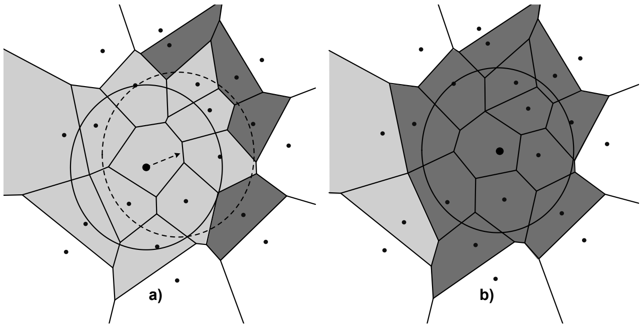

Figure 2: Neighbor discovery: a) Neighbors about to be discovered shown in dark gray. b) Neighbors no longer within moving players AOI shown in light gray.

send by neighbors.

3.3 Generation of player movement

Our implementation of VAST uses OverSim [2], an overlay network simulation framework for the OMNeT++ simulation environment [11]. Apart from an overlay layer, which encapsulates VASTs functionality, we also built an application that sits on top of VAST. This application generates movement data for each player during a simulation run. Movement data can be based on various movement models, in the following however we only look at two of them.

Most MMOGs focus on two different target groups, namely players looking for a singleplayer-like experience and players engaging in teamplay. There is a large number of players, who despite most MMOGs interactive nature, enjoy wandering through the game world on their own. Interaction with other players happens every now and then, but basically they are following their own goals, much like in a single player game.

The main focus though, are players wandering through the game world together in small to medium sized groups, solving quests and fighting monsters or even other groups together.

Both of these scenarios are supported via two movement models called *Random Waypoint* and *Group-based Random Waypoint*. We employ basic flocking techniques [7], in order to achieve realistic group movement patterns as proposed in [10].

In *Random Waypoint* mode players are evenly distributed within the game world and basically pick a random target. Upon arrival at their current target they simply choose another one and continue to walk around the game world. There is nearly no interaction between players, except for avoiding collisions when getting close to each other. In *Random Waypoint* mode players neighbors lists should be relatively sparse, as they do not get to close to each other most of the time.

In *Group-based Random Waypoint* mode group starting positions are distributed over the whole game world and players joining consecutively are put together in groups of adjustable size. Like in *Random Waypoint* mode each group chooses a random target within the game world and tries to reach it, while individual players try to avoid collisions within the group and at the same time try to stay close to the center of the group. The groups target destination in our implementation is distributed rather simple among players belonging to the same group. Current targets of every group are globally available and players are regularly checking for an updated target. In *Group-based Random Waypoint* mode there is a lot of interaction between players as their movement is based on nearest neighbors positions. The number of neighbors for each player should roughly match the predefined group size depending on wether he is closer to the center or the border of the group.

4. EVALUATION

4.1 Simulation setup

Evaluation starts with a look at the bandwidth requirements of VAST. Then overlay consistency over time is evaluated for two different scenarios. In the first scenario a fixed number of 500 players randomly wandering the game world, each player on its own, is simulated via our *Random Waypoint* model. This scenario helps to determine minimum bandwidth requirements but does just provide a very simple player movement. In the second scenario the *Group-based Random Waypoint* model is used to simulate 500 players wandering the game world in groups, by employing our waypoint driven flocking algorithm.

Bandwidth requirements are measured in average and maximum needed kb/s per player. Overlay consistency is measured by calculating a global view of the overlay network and comparing it to the local views of the players.

We choose an AOI-size of 60 meters around each player and an average movement speed of 6 meters per second, which is the mean walking speed of a human. These values were used during all simulation runs. Where group roaming was used, the average number of players per group was set to 12. The simulated game world was a square area 10.000 meters wide and long.

4.2 Bandwidth requirements

	average bandwidth per player	maximum bandwidth per player
Random Waypoint	0,73 kb/s	6 kb/s
Group-based RW	1,26 kb/s	8 kb/s

Table 1: Average and maximum bandwidth requirements for the two movement models.

The bandwidth measurements of VAST have been determined with a series of ten simulation runs, with a constant number of players for each run. After each run the average bandwidth requirements for all players were calculated. Then the average bandwidth requirements of the ten simulations were calculated accordingly. Table 1 lists the obtained results for 500 simulated players.

Growing player numbers for different series of simulation runs barely affected the obtained bandwidth measurements, as long as players were sparsely distributed over the game world. Sparse distribution for a growing number of players was achieved by enlarging the game world size accordingly. Thus VAST seems to scale quite well with growing numbers of players. This is due to the fact, that the number of average neighbors per player is independent of the overall

	average number of neighbors per player
Random Waypoint	5,3
Group-based RW	8,4

Table 2: Average number of neighbors for the two movement models.

number of players, provided the game world is big enough. Table 2 lists these numbers for our two movement models.

VASTs bandwidth requirements are comparatively low and can be satisfied by a common DSL-connection. In a real world application bandwidth requirements may be slightly higher due to various player actions that could be implemented. The greater part of bandwidth is needed for movement updates and network traffic for maintaining the overlays structure though and that is exactly what we measured in our simulations. Also there is still enough bandwidth left even on a common DSL-connection for a lot more events. Since all neighbors maintain direct connections, additional traffic should not stress the overlay too much, because most messages are just exchanged between two neighbors and there is no need to forward them over long distances via other players.

Table 1 also lists maximum bandwith requirements that occurred during simulation. These peaks mainly occur during simulation start-up, when many players join the overlay network and many new neighbors are exchanged. Later on only new found neighbors are exchanged between players and actual bandwith requirements per second quickly draw near the average bandwith shown in table 1.

Though VASTs bandwidth usage appears to be nearly independent of growing numbers of players, table 1 shows an increase in bandwidth usage from *Random Waypoint* mode roaming to *Group-based Random Waypoint* mode. This is also reflected in table 2. The average number of neighbors starts to increase, when several players get close to each other and gather around a common spot. With an increased number of average neighbors per player, bandwidth usage has to rise in order to accommodate maintenance traffic needed to preserve the overlays structure. Also there are obviously more movement updates exchanged between players.

4.3 Overlay consistency

Overlay consistency is measured by looking at faulty entries in players neighbors lists. We used a fixed number of players in our two different movement models and performed an overlay integrity check every five minutes.

The overlay integrity check takes a global list of all players, available through OverSim, and iterates through them, calculating a global voronoi diagram for each player. Then players local voronoi diagrams are checked against the according global voronoi diagram and missing or superfluous entries in their neighbors lists are counted. By averaging over all players and simulation runs, we get the mean number of missing or superfluous neighbors per player at a given time. Figure 3 shows the results for our first movement model.

The overlay consistency was measured over a period of 6 hours. Figure 3 lists the average number of missing and superfluous entries within players neighbors lists over time.

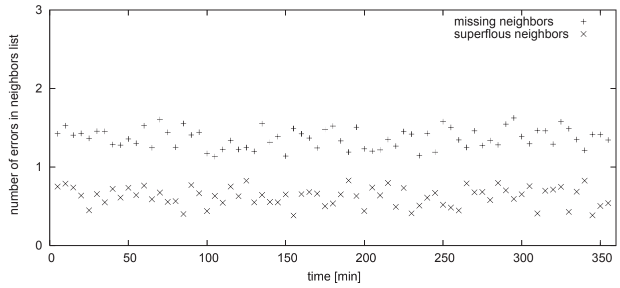

Figure 3: VASTs overlay consistency for players in *Random Waypoint* mode.

Consistency in our *Random Waypoint* model is reasonable good, considering the fact that the overlay integrity check is not expected to be absolutely exact. We ignore messages not yet delivered, when comparing local with global voronoi diagrams. In cases, where a player just changed the position but the messages containing the position updates are still transported to his neighbors, it is possible that the local voronoi diagrams lags behind the global voronoi, thus generating a few faulty entries in the neighbors lists. As seen in figure 3 faulty neighbors list entries are low at all times and imply a stable and consistent overlay network.

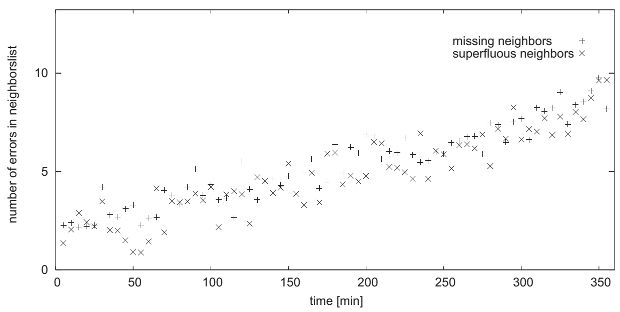

Figure 4: VASTs overlay consistency for players in *Group-based Random Waypoint* mode.

Figure 4 displays how the overlay consistency develops over a period of 6 hours for our *Group-based Random Waypoint* model, with groups of 12 players each. In contrast to the *Random Waypoint* model the number of errors in the *Group-based Random Waypoint* model is constantly growing over time, up to the point where most of the entries in each players neighbors list have to be faulty. Further examination of the groups reveals that single players are cut off from the group over time and are unable to recover any connections to other players within the overlay. This leads to a completely disjointed overlay after 6 hours, that has fallen apart into several separated pieces.

5. REASONS FOR OVERLAY INCONSISTENCIES

Figure 5 shows which problems arise, when players get close to each other so that movement speed considerably exceeds mean player distance among one another. The player pictured by a square moves alongside the arrow, thus not en-

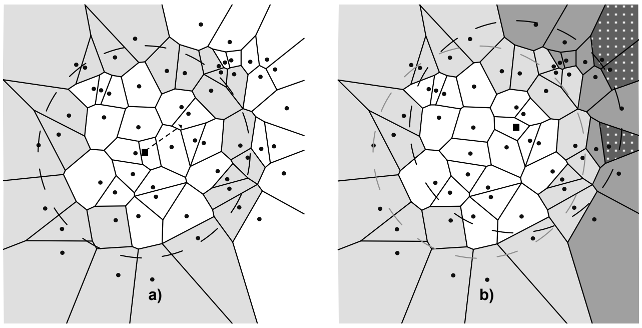

Figure 5: Undiscovered neighbors: a) Boundary neighbors of moving player (square) in light gray. b) New neighbors about to be discovered in dark gray and missed neighbors in dark dotted gray.

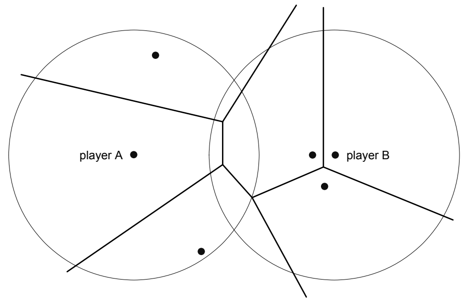

Figure 6: Non bilateral neighborhood: Player A is a boundary neighbor of player B, but player B is no neighbor of player A.

tering his neighbors region but completely skipping it. His boundary neighbors are colored in light gray (Figure 5a). Upon movement of the square marked player his boundary neighbors query their enclosing neighbors and check whether their voronoi regions overlap with the moving players AOI thus becoming new neighbors of the moving player (Figure 5b). New found neighbors are marked in dark gray. But there are three new neighbors not discovered in this case, marked in dark dotted gray. This is because the players are located very close to each other and the moving players AOI-border is skipping other players entire voronoi regions. Unfortunately these neighbors remain lost as long as they remain within the moving players AOI, because in subsequent test against the moving players updated AOI these neighbors are assumed to be known by the moving player. As only new found neighbors are exchanged to keep bandwith requirements low, these missing neighbors are probably never added to the moving players neighbors list again.

This explains the bad results of our second overlay consistency measurements, when using *Group-based Random Waypoint*. Due to our group management players disconnected from great parts of their group are still able to follow the groups path, as the groups current target destinations are globally available. While moving from the border to the center of a group a player can easily lose connections to players at the border of his AOI, which are not recovered because of the lost players still following the groups path and possibly staying within the others players AOI for a long time. This is what is likely to happen in a real world application, when a player loses several of his neighbors but still tries to follow those he still sees.

Another source of inconsistencies are non bilateral neighborhood relationships. Enclosing neighbors are obviously always bilateral, because when two players regions share a commen edge this has to apply to both players. But as figure 6 illustrates this is not always true for boundary or *normal* neighbors, which do not necessarily have to be bilateral. In this case player B would connect to player A and add player A to his neighbors list. But player A would never send position updates to player B, because player B is not in player A's neighbors list. Updates send from player B to player A are just dropped, because player A has no interest in the position of player B. Player B then removes player A from his neighbors list, assuming player A has moved away from him or left the overlay.

This issue also just affects the boundary and *normal* neighbors but not the enclosing neighbors. In *Random Waypoint* mode overlay consistency mainly depends on enclosing neighborhood relationships, because players are wide spread throughout the game world and their voronoi regions are relatively big compared to their AOIs. So there is not so much overlapping of players AOIs as there are enclosing neighborhoods through shared voronoi edges.

This explains why the overlay remains stable with players in *Random Waypoint* mode, while it starts to collapse with players in *Group-based Random Waypoint* mode.

6. CONCLUSION

In this paper, we presented the basic functionality of VAST, a fully-distributed peer-to-peer protocol, designed to handle event messages in MMOGs. VAST was proposed by Shun-Yun Hu and Guan-Ming Liao in *Scalable Peer-to-Peer Networked Virtual Environment*. We evaluated an OverSim-based implementation of VAST with regard to bandwidth requirements and overlay consistency.

VAST shows optimal end-to-end delay results, because all communication between players takes place through direct connections, thus employing only one overlay hop at all times. This is achieved by relying on players true AOI instead of fixed sized regions of the game world. Therefore unnecessary network overhead in VAST is low.

Neighborhood discovery in VAST is handled by voronoi diagrams. Players need to constantly rebuild their local voronoi diagram on incoming messages from their neighbors. Hence it becomes vital that a fast algorithm for constructing voronoi diagrams is employed, in order to accomplish a low latency when processing messages send by neighbors. The fastest known algorithms manage to build voronoi diagrams in O(n log n) time. Construction of voronoi diagrams is computing intensive though and latency can get problematic on slow PCs in cases where there are many neighbors.

VASTs bandwidth requirements were measured in different scenarios and are overall good and easily satisfiable by a common DSL-connection. Bandwidth usage increases when several players gather in a small area, but is barely influenced by the overall number of players. At least as long as the game world is big enough to allow for the players to be sparsely distributed over the whole area. Otherwise it is inevitable that the player density per area and consequently the bandwidth usage increases.

Unfortunately serious problems arise regarding VASTs consistency. Two different scenarios have been simulated, where players wander around the game world on their own or in groups, in each case following random waypoints. We called these scenarios *Random Waypoint* and *Group-based Random Waypoint*. In *Random Waypoint* mode, where players are sparsely spread throughout the game world and overlay consistency mainly depends on players enclosing neighbors, consistency is very good and remains stable. In *Group-based Random Waypoint* mode however, players start to lose neighbors over time, up to the point where the overlay is separated into several isolated parts.

We pointed out two major flaws in VASTs design, which cause overlay inconsistencies. Neighbors can get lost, when groups of players get close to each other and movement speed considerably exceeds mean player distance among one another. The other problem is associated with VASTs handling of boundary neighbors. Boundary neighborhoods are, contrary to enclosing neighborhoods, not necessarily bilateral. This can lead to situations, where one player expects to obtain movement updates from another player, who is not even aware of the first player.

After all VAST performs quite good in *Random Waypoint* mode. Especially the enclosing neighbors approach looks promising for preserving overlay consistency in cases, where players are sparsely spread around the game world and connections beyond players AOIs need to be sustained, in order to keep the overlay topology connected. Considering the fact, that consistency issues arise in *Group-based Random Waypoint* mode, VAST seems not yet applicable in real world scenarios. We however are confident, that these issues can be solved and want to try to enhance VAST to perform equally well in group-based scenarios as it already does in *Random Waypoint* mode.

7. FUTURE WORK

As mentioned in section 3.3, a key feature in most MMOGs is to form groups of players that play together for a certain time. This groups build a fixed set of players that will most probably be neighbors in the overlay for a longer time. We believe, that knowledge of these group could be used to improve the stability of the overlay and avoid the situations described in section 5. To test this hypothesis we are implementing a group management extension for VAST.

As shown in section 5, players that have lost a number of their neighbors have only a slim possibility to recover from this situation. We want to extend VAST with a backup feature that allows players to retrieve lost neighbors.

We will enhance neighbor discovery, so that a moving players updated AOI is checked against all neighbors of the moving players boundary neighbors, instead of just looking at his boundaries enclosing neighbors. Alternatively intermediate steps could be added when calculating the voronoi diagram for fast moving nodes. The first solution will increase bandwidth requirements while the second solution will increase the per message latency. We will investigate which solution provides better results regarding stability and scalability.

Lastly it looks promising for each player to not only keep a list of his neighbors, but to also keep a list of other players, which are interested in his position. That way we will try to eliminate the non bilateral boundary neighbors problem. This also opens up the option to exchange AOI-sizes among players, thus making it possible for a players AOI to adept to his surroundings. We believe in doing so bandwidth requirements can be dropped further in certain situations, where players just have a limited view of the game world. This can happen when large objects block a bigger part of a players view or in areas where it is foggy for instance.

8. REFERENCES

[1] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, 1991.

[2] I. Baumgart, B. Heep, and S. Krause. Oversim: A flexible overlay network simulation framework. In *Proceedings of 10th IEEE Global Internet Symposium (GI '07) in conjunction with IEEE INFOCOM 2007, Anchorage, AK, USA*, May 2007.

[3] S. Fortune. A sweepline algorithm for voronoi diagrams. In *SCG '86: Proceedings of the second annual symposium on Computational geometry*, pages 313–322, New York, NY, USA, 1986. ACM Press.

[4] S.-Y. Hu and G.-M. Liao. Scalable peer-to-peer networked virtual environment. In *NetGames '04: Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*, pages 129–133, New York, NY, USA, 2004. ACM Press.

[5] T. Iimura, H. Hazeyama, and Y. Kadobayashi. Zoned federation of game servers: a peer-to-peer approach to scalable multi-player online games. In *NetGames '04: Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*, pages 116–120, New York, NY, USA, 2004. ACM Press.

[6] B. Knutsson, H. Lu, W. Xu, and B. Hopkins. Peer-to-peer support for massively multiplayer games. In *INFOCOM*, March 2004.

[7] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, New York, NY, USA, 1987. ACM Press.

[8] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. LNCS2218:329–350, 2001.

[9] A. I. T. Rowstron, A.-M. Kermarrec, M. Castro, and P. Druschel. Scribe: The design of a large-scale event notification infrastructure. In *NGC '01: Proceedings of the Third International COST264 Workshop on Networked Group Communication*, pages 30–43, London, UK, 2001. Springer-Verlag.

[10] J. H. Seu, B.-K. Song, and H. S. Kim. Simulation of artificial life model in game space. In *Artificial Intelligence and Simulation*, pages 179–187, Heidelberg, Germany, 2005. Springer Berlin.

[11] A. Varga. Omnet++ 3.2 user manual. http://www.omnetpp.org/doc/manual/usman.html, May 2007. accessed on 05-24-2007.

[12] S. Yamamoto, Y. Murata, K. Yasumoto, and M. Ito. A distributed event delivery method with load balancing for mmorpg. In *NetGames '05: Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*, pages 1–8, New York, NY, USA, 2005. ACM Press.