AOI-Cast Strategies for P2P Massively Multiplayer Online Games

Luca Genovali
IMT, Institutions, Market, Technologies
Piazza San Ponziano 6, Lucca, Italy
Email: l.genovali@imtlucca.it

Laura Ricci
Dip. di Informatica, Università degli Studi di Pisa
Largo Bruno Pontecorvo 3, Pisa, Italy
Email: ricci@di.unipi.it

***Abstract*—This paper presents a $P2P$ support for *Massively Multiplayer On line Games*. The $P2P$ overlay is defined by considering a *Voronoi diagram* where the sites correspond to the peers and a link connects Voronoi neigbours. An *AOI-cast mechanism* is defined to notify events generated by a peer to peers located in its *Area of Interest*. A routing algorithm exploiting the properties of the Voronoi graph has been defined as well. *AOI-cast* has been evaluated through a set of simulations developed by *Peersim*.**

***Keywords*: Peer-to-peer, multiplayer, routing, network**

I. Introduction

The development of Distributed Virtual Environments, like *Massive Multiplayer Online Games, MMOG* [3], [6], [10] on *P2P architectures* is currently an active challenging research area. As a matter of fact, commercial *MMOG* are generally developed on *client server architectures* which suffer of evident scalability problems, due to the huge amount of players a single server has to manage.

The definition of a scalable communication support is a basic issue for the wide diffusion of these applications in a $P2P$ environment. Several proposals exploit the concept of *Area of Interest, AOI,* to define this support [1], [10]. The *Area of Interest* of a player P is a region of the virtual world surrounding P and such that P is aware only of other players and passive objects located in this area. The implementation of the AOI requires the definition of a *highly dynamic P2P overlay*. As a matter of fact, since the positions of the players change continuously, new players may enter the AOI of a peer while others leave it. Since a peer should be dynamically connected with other players in its *AOI*, the structure of the overlay is highly dynamic and proper mechanisms are required to guarantee its connectivity.

A recent proposal [10] exploits *Voronoi diagrams* [9] to define a dynamic overlay where each peer is connected to its neighbours in the *Voronoi diagram*. To implement *Areas of Interest* on Voronoi based overlays an *AOI-cast* [12] mechanism, i.e. an *application level multicast* constrained within *AOI* boundaries, should be defined. A straightforward implementation of *AOI-cast* exploits routing on Voronoi links to reach any peer of the AOI. The main problem of this approach is the *high latency* due to the multiple routing hops through intermediate peers. A high latency may be experimented especially in *crowding scenarios*, i.e. when a large number of peers is present in the AOI.

An alternative solution adds to the Voronoi overlay a set of direct links between a peer and any other peer in its AOI. Voronoi links are preserved to guarantee the connectivity of the overlay when the AOI of a peer is empty. The main drawback of this approach is that peers characterized by a low communication bandwidth cannot support an high amount of connections, especially in *crowding scenarios*.

This paper presents an intermediate solution, where a peer exploits direct connections with a *subset of the peers in its AOI* and a proper routing algorithm is defined to route notifications to any other peer. By exploiting the properties of the *Voronoi diagrams*, our approach minimizes the amount of information required by routing. Furthermore, the discovery of new peers exploits the basic mechanism of event notification and does not require to explicitly query neighbours to discover new peers entering the AOI.

Sect. II describes related work. Voronoi based approaches are presented more deeply in Sect.III. Our approach is defined in Sect. IV, while the routing algorithm is described in Sect. V. Experimental results are discussed in Sect. VI. Finally Sect.VII presents some conclusions.

II. Related Work

Solipsis [6] and *Apolo* [14] define a proper dynamic overlay for *MMOG*. In *Solipsis* nodes are self-organized in a pure peer-to-peer network, in which relationships between nodes depend upon virtual proximity. In *Apolo* [14] each node connects to the closest neighbours in each of the four quadrants. To implement *AOI-cast* each node sends a message to its closest neigbours and the message is then forwarded until it reaches the border of the AOI. In this way each node defines direct links with its four neighbours only. As a counterpart, the number of hops required to reach a peer belonging to the AOI may be high. *Mopar* [11] considers an alternative approach exploiting a *Distributed Hash Table* to assign passive objects to the peers. Since the approaches based on *Voronoi graphs* are more strictly related to our approach, they will be analysed more deeply in the next section.

III. Voronoi Based P2P Overlays for MMOG

The definition of a P2P overlay for *MMOG* based upon *Voronoi Diagrams* [9] has been investigated in [1], [13], [12],

978-1-4244-2309-5/09/$25.00 ©2009 IEEE

[10].

In a *Voronoi based* approach, the position of each peer in the virtual world is exploited to define a *Voronoi tessellation* of the virtual world. Given *n sites* corresponding to the peers, a *Voronoi tessellation* partitions the virtual world into *n areas* such that the area corresponding to a site n includes all the points which *are nearer* to n with respect to any other site. Two sites are *Voronoi neighbours* iff the borders of their areas overlap. The connected graph defined by linking neighbour sites corresponds to the *Delaunay Triangulation* associated to the *Voronoi tessellation*. A $P2P$ overlay is defined by connecting peers whose sites are Voronoi neighbours. In the following, the links of this overlay will be referred as *Voronoi links*. The problem of maintaining the *Voronoi* structure of the overlay in a dynamic $P2P$ environment, like a *MMOG*, where the positions of the peers change continously and no centralized coordination entity does exist, is a challenging issue.

In a *MMOG* any event generated by a peer should be notified to any other peer in its *AOI*. Even if a peer can generate distinct kind of events, in the following we will consider the most frequent one, i.e. *its movement*, which should be periodically notified through a *heartbeat message* to the other peers located in its AOI. This notification may be implemented through a *AOI-cast* mechanism [12], i.e. an application level multicast constrained within the boundary of the area of interest. *Flooding* the heartbeat through the *Voronoi links* generates a large amount of redundant messages and presents evident scalability problems. A more refined approach dynamically computes a *spanning tree* on the *Voronoi links* including all the peers of the AOI and exploits this tree to notify the heartbeat. Both solutions are based on *forwarding*, i.e. any heartbeat is routed to the peers in the AOI through a sequence of intermediate peers. An obvious drawback of these solutions is the high latency in the delivery of an event, especially in *crowding scenarios*, i.e. when a set of peers lie close to each other in the *virtual world* and their *AOI* overlap. In this case *several routing hops* may be required to notify an event due to the large amount of peers located in the *AOI*. The resulting *latency* may be not tolerable in *MMOG* and may compromise the interactivity of the application. On the other hand, since the number of *Voronoi neighbours* is 6 on average [9], in this solutions a peer manages a small number of connections.

An alternative solution [10] defines *direct links* between a peer and any other one in its AOI. The resulting overlay includes these links besides the Voronoi ones, which have to be maintained to guarantee the connectivity of the overlay. This solution minimizes the latency, but increases the number of connections of each peer. In a crowding scenario like the one previously described, a peer should manage a large number of connections, since a large amount of peers are located in its AOI.

The approach of *dynamically enlarging or shrinking* the size of the AOI [10] according to the bandwidth of the peers is not fair, because the size of the AOI depends upon the semantics of the application and players with a larger AOI could be favored.

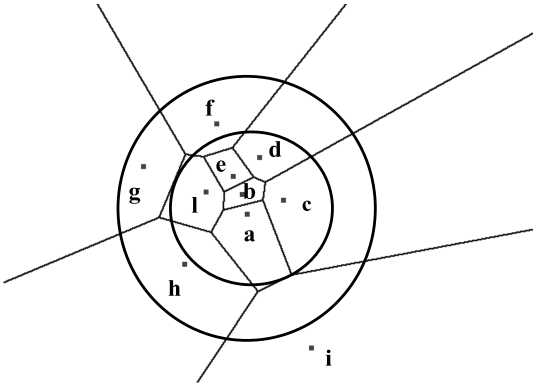

Fig. 1. Internal and Peripheral Areas of Interest

We propose an intermediate solution, which defines direct connections between a peer P and a *subset of the peers in its AOI*, i.e. the peers nearer to P, and exploit *forwarding* to reach any other peer of the AOI. Following sections will describe our approach as well as a set of mechanisms to increase the overall scalability.

IV. AOI-cast: Direct Links and Forwarding

This section describes an *AOI-cast* exploiting both *direct links* and *forwarding*.

The *Area of Interest* of each peer P is partitioned into two areas, the *Internal Area of Interest* of P, $IAOI(P)$ and the *Peripheral Area of Interests*, $PAOI(P)$, $PAOI(P) = AOI(P) - IAOI(P)$. In the following, we will consider circular areas centered at P, but our approach can be generalized to different shaped areas.

The $P2P$ overlay includes, besides the *Voronoi links*, a link between P and any peer located in $IAOI(P)$. Each heartbeat is periodically sent by P through these links, afterward it is forwarded to any peer located $PAOI(P)$.

Fig.1 shows the $IAOI$ of a peer a which is delimited by the internal circumference, while its $PAOI$ overlaps the *annulus* between the circumferences. The corresponding overlay includes direct links between a, and rs. b,c,d,e,l, while *forwarding* is exploited to reach g, f and h. Notice that since a and e are not *Voronoi neighbours*, the direct link connecting them is not a *Voronoi link*. Furthermore, it is worth noticing that a should send its heartbeats to i as well, even if it is located outside its area of interest because any *Voronoi Neighbour* act as 'beacon node' that inform a about peers approaching from farthest positions. Network connectivity is guaranteed by these nodes, *even when the AOI of a peer is empty*.

In the following, we will assume that the width of the annulus of $AOI(P)$ is larger than 0, i.e. the border of $IAOI(P)$ and of $PAOI(P)$ do not overlap. Furthermore, this width is chosen so that any peer Q entering $AOI(P)$ sends an heartbeat when it is located in $PAOI(P)$, before entering $IAOI(P)$. To compute the *minimal width* of the annulus which guarantees

```
$Q$ ::
receive $hb(P)$ from $S$
if $Q \in AOI(P)$
    $PN = \{n : n \in VoronoiNeighbours(Q), n \in PAOI(P)\} - \{S\}$
    if $(PN \neq \emptyset)$
        foreach $n \in PN$
            if $is\_father(P, Q, n)$ send $hb(P)$ to $n$
else
    $SN = VoronoiNeighbours(P) \bigcap VoronoiNeigbours(Q)$
    foreach $n \in SN$
        if $n \notin AOI(P)$ send $hb(P)$ to $n$
```

Fig. 2. AOI-cast: High Level Specification

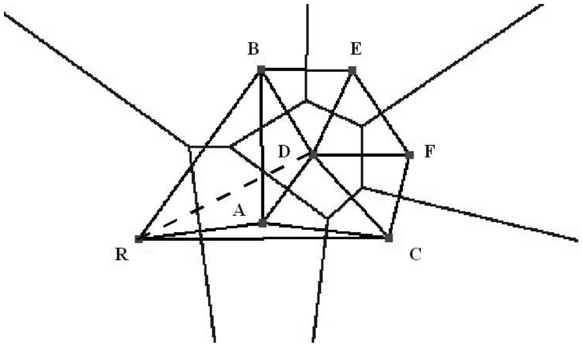

Fig. 3. Compass Routing: Computation of the Spanning Tree

the previous condition, both the *maximum speed* of a peer and the *frequency of heartbeat notifications* should be considered.

The peer P which generates an heartbeat sends its to all its Voronoi neighbours and to any peer in its $IAOI$. The forwarding of the heartbeat is started by the peers located 'at the border' of its $IAOI$, i.e. the *boundary peers*. Then, any peer located in $PAOI(P)$ carries on the forwarding.

Fig. 2 shows an high level specification of the procedure executed by a peer Q when it receives from peer S an heartbeat generated by peer P. The forwarding phase exploits a *routing algorithm* based upon the construction of a *spanning tree* rooted at P. We suppose that a boolean function $is\_parent(P, Q, n)$ does exist which returns true iff Q is the parent of peer n in the spanning tree rooted at P. Section V will define this function by exploiting information local at each peer only.

A peer Q is a *boundary peer* for peer P iff one of its Voronoi neighbours is located in $PAOI(P)$. When Q receives an heartbeat from P it checks whether it is a *boundary peer* for P. In this case, it exploits $is\_parent(P, Q, n)$ to check if its Voronoi neighbours belonging to $PAOI(P)$ are also its children in the spanning tree rooted at P. In this case, it forwards the heartbeat to each child. Forwarding is carried on by each peer belonging to $PAOI(P)$. It is worth noticing that not all the peer receiving the heartbeats are involved in the forwarding phase. This is the case of peer b in Fig. 1 whose neighbours are all located in $IAOI(a)$.

If Q does not belong to $AOI(P)$, i.e. the else branch in Fig.2, it should be a *Voronoi neighbour* of P located outside $AOI(P)$. In this case, Q notifies the heartbeat to each peer V which is Voronoi neighbour for both P and Q and that does not belong to $AOI(P)$. Notice that V may not know P because it is approaching P from a further position with respect to that of Q. In this case, Q 'puts in touch' P and V. These notifications are critical, because they guarantee network connectivity when the area of interest of a peer is empty. Note that the same hertbeat can be notified to a peer approaching P by more peers, but in this case, a certain degree of *redundancy* may decrease the probability of overlay disconnections. The robustness of the application may be increased by introducing further redundancy. For instance, a heartbeat generated by P and received by a peer Q located outside P Area of Interest may be propagated to all the Voronoi neighbours of Q, even to those that are not neighbours of P. In this case, an abrupt crash of Q will not disconnect the network because a larger set of peers knows the position of P.

It is worth noticing that in our approach the heartbeat notification mechanism naturally supports the discovery of new neighbours. Each peer Q approaching P, must first keep in touch with a Voronoi neighbour V of P or with a peer located in $AOI(P)$ (recall the width of the annulus is different from 0). V simply propagates an heartbeat received from P to Q. In this way, Q becomes acquainted with P. Notice that this differs from the approach proposed in [10], where each peer acquires knowledge of new approaching peers by *explicitly querying* its boundary neighbours.

Finally, it is worth noticing that *AOI-cast* can be further optimized by exploiting a set of *aggregation techniques*. As a matter of fact, if Q is the parent of V with respect to a set of spanning trees routed at different peers, it may *aggregate* the heartbeats sent by these peers in a single message which is periodically sent.

V. Routing in Voronoi Overlays

The scalability of the *AOI-cast* strategy described in Sect.IV greatly depends upon the efficiency of the *routing algorithm* implementing the forwarding phase. This issue is especially important when a *crowding* occurs in the $PAOI$.

This section describe *compass routing* [8], [7], a routing algorithm which exploits the properties of *Delaunay Triangulations* to minimize the information each routing step requires.

Compass routing is based upon the following observation. Consider a connected graph G and assume of being located at a node n of G with the goal to reach a destination node d. [8] shows that the best strategy looks at the edges incident in n and chooses the edge whose slope is minimal with respect to the segment connecting n and d. [8] shows that while *compass routing* is not cycle free for general graphs, it can always find a finite path between two nodes of a *Delaunay Triangulation*.

The original formulation of *Compass Routing* makes it possible to discover a path from a node n *towards* a node r. In our case, it is necessary *to reverse* any path defined by the original algorithm, to build a *spanning tree* rooted at the node generating the heartbeat.

A simple solution, based upon the original definition of compass routing, requires that a node n knows, not only its Voronoi neighbours, but also those that are distant two Voronoi hops, i.e. the neighbours of its Voronoi neighbours. Since n is the parent of a node v iff v chooses n as its parent by compass routing, n must be aware of the position of any neighbour of v to compute the reverse path.

An optimization of this procedure may be defined by considering a basic property of the *Delaunay triangulation*. We recall that any *Delaunay Triangulation* satisfy the *empty circle condition* which states that the circumcircle of each triangle belonging to the triangulation is empty, i.e. it does not contain vertexes besides those that define it. Consider now the *Voronoi Diagram* and the corresponding *Delaunay Triangulation* in Fig. 3. Let us suppose that R is the peer generating the heartbeat, i.e. the root of the spanning tree and let us consider node A which receives the heartbeat directly from the root. A should decide whether it is the parent of node D in the spanning tree. A may apply *compass routing* by considering only the triangles ABD and ACD and by comparing only the slopes of the edges AD, BD, and DC with respect to the segment RD. As a matter of fact, the empty circle property guarantees that other edges incident in D cannot intersect these triangles, hence their slope with respect to the segment RD is larger. Hence, A is the parent of D iff the angle $\angle ADR$ is smaller than $\angle BDR$ and $\angle CDR$.

In this case, each node only needs its own coordinates, the coordinates of its Voronoi neigbours and those of the root of the spanning tree to determine its children in the tree. This minimizes the amount of information to implement *compass routing* and the number of messages exchanged through the overlay. For this reason, our approach is more scalable with respect to [12] that requires a larger amount of information.

It is important to note that in a distributed environment the local views of the virtual world at each peer may be inconsistent, due to network delay, packet loss. For this reason, different notifications of the same heartbeat may reach a peer, while some notification may not reach a peer at all. The discussions of the techniques reducing these inconsistencies (for instance dead reckoning)[1], is outside the scope of this paper.

The following theorem shows that in the case of circular areas of interest, only peers belonging to the AOI of a peer P should be considered when building a spanning tree including all the peers in $AOI(P)$.

Theorem 1: If $AOI(P)$ is a circular area centered at P, compass routing can compute a spanning tree rooted at P which includes *all and only the nodes* $\in AOI(P)$.

Proof: [8] shows that each step of compass routing on Delaunay graphs decreases the distance to the destination. Since the construction of the spanning tree reverses any path computed by compass routing, the distance from the root increases at each routing step.

Consider a peer $V \notin AOI(P)$. Since the distance of this peer from the root P is larger than the radius of the $AOI(P)$, V cannot be the parent of a peer $Q \in AOI(P)$, otherwise the distance from the root should decrease when passing from V to Q. ■

Note that the theorem may be not valid for different shaped area of interest. As shown in [2], if rectangular or squared areas are considered, some path of the spanning tree may zig zag around the borders of the Area of Interest.

The theorem guarantees that only peers belonging to the Area of Interest are to be considered when computing the spanning tree. Furthermore, it is worth noticing that the spanning tree we compute includes a set of *'shortcuts'* connecting the root to its boundary peers. The result is valid also in this case, because any peer belonging to the *PAOI* should be a child of a boundary peer or of another peer in the *PAOI*.

VI. Experimental Results

This section describes a set of preliminary evaluations of our approach. The results have been obtained by a set of simulations implemented through *Peersim* [15], a *modular* and *highly scalable* simulator supporting the simulation of $P2P$ systems including a large number of peers and that supports the evaluation of new overlay topologies and protocols.

We consider a 800×600 bidimensional virtual world. At the beginning of the simulation the peers are randomly distributed in a uniform way within the virtual world. Each peer may change its direction at random at each simulation cycle and it turns back if hits the border of the grid. The *speed* of a peer determines the space it covers at each simulation cycle and it is measured in pixels. It is possible to change the speed of a peer from 1 to 12 pixels. *Peersim* enables to configure each experiment through a configuration file. In our experiments, it is possible to define the *number*, the *speed* and the *probability of a change of direction* of the peers. Furthermore, the radius of the *Internal Area of Interest* may be defined within the range $[0, 50]$.

The goal of the first set of experiments is to investigate some topological properties of the overlay when the length of the radius of the *Internal Area of Interest* increases from 0 to 50. In all the charts *nigh* corresponds to the length of the radius equal to 0. The top, rs. the bottom chart in Fig. 4 show the *average*, rs. the *maximum number*, of links managed by 1000 peers during 1500 simulation cycles. In the experiment, the peer speed is constant and the radius of the *Internal Area of Interest* varies from 0, i.e. each peer is connected to its Voronoi neighbours only, to 50.

The experimental results confirm the theoretical one which states that the average number of neighbour is 6, when only Voronoi links are considered. Since we consider a uniform distribution, the average number of links is roughly constant across different simulation cycles. The bottom chart shows the *maximum number of links* at each cycle when considering all the peers. The chart reports the measures corresponding to length 0,rs. 40 and 50 of the radius, while those corresponding to length 10, rs. 20 and 30 are not shown because they roughly overlap those reported for length 0. The results show that,

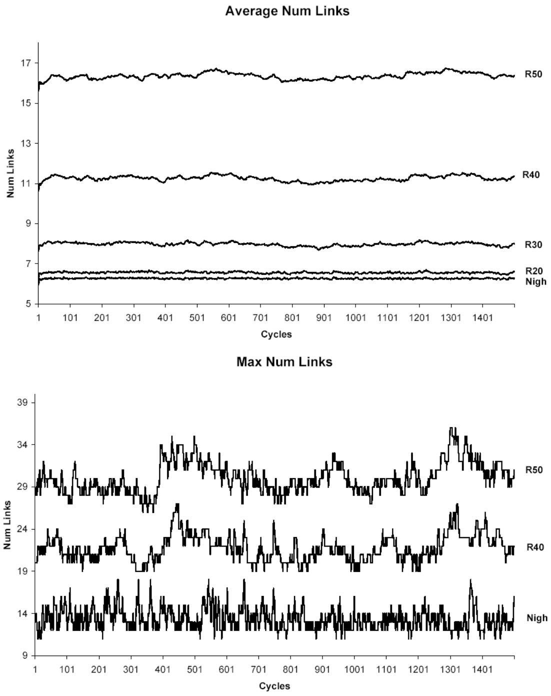

Fig. 4. Topological Properties of the Overlay

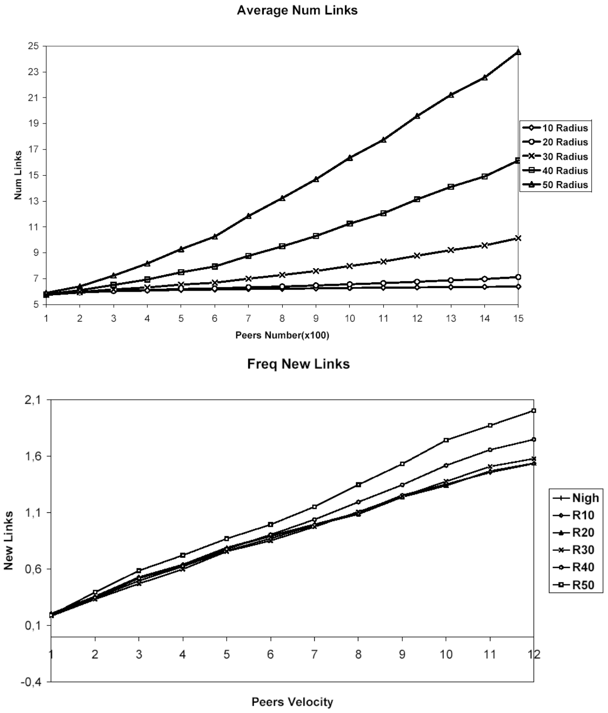

Fig. 5. Scalability

when the length of the radius exceeds 40, a large amount of connections should be supported by each peer.

The second set of experiments evaluates the *scalability* of the overlay as the number of peers increases. The top chart in Fig.5 reports the average number of links, for different length of the radius and when the number of peers increases from 100 to 1500. Even if the number of links increases with the number of peers, the results show a good overlay scalability when the length of the radius is smaller than 30. As a matter of fact, each peer has less than 10 links on the average, for a radius smaller than 30. This result shows the feasibility of a trade off solution exploiting only Voronoi links in the general case and an *Internal Area of Interest* characterized by a small radius in crowding scenarios. As a matter of fact, while a small radius corresponds to a reasonable number of links, it may constrain the number of routing steps with respect to the solution including Voronoi links only.

The final set of experiments tests the *degree of variance* of the overlay, as a function of peers speed by increasing it from 1 to 12, and by considering different radius. The bottom chart in Fig.5 shows the average number of *new links* created by each peer at each simulation cycle. The results show that the length of the radius has a modest impact on the overlay variance.

VII. Conclusion

This paper defines *AOI-cast*, a scalable mechanism to notify events within the *Area of Interest* of each player of a *P2P MMOG*. Our proposal may be extended in several directions. First of all, a set of mechanisms to support *passive objects* will be defined. We are also evaluating *hybrid P2P architectures*, where each peer may have a different role in the $P2P$ network according to its computational power.

References

[1] L.Ricci, A. Salvadori *Nomad: Virtual Environments on P2P Voronoi Overlays*, 1th Int. Work. on Peer to Peer Networks (PPN 2007), Vilamoura, LNCS, Vol. 4803, November 2007.

[2] M.Albano, M. Baldanzi, R. Baraglia, L. Ricci *VoRaQue: Range Queries on Voronoi Overlays* 13th IEEE ISCC, Marrakesh, July 2008.

[3] L. Genovali, L.Ricci *JaDE: A JXTA Support for Distributed Virtual Environments* 13th IEEE ISCC, Marrakesh, July 2008.

[4] F.Baiardi, A. Bonotti, L.Genovali, L.Ricci. *A publish subscribe support for networked multiplayer games*, EuroIMSA 2007, Chamonix, March 2007.

[5] A.Bonotti, L.Genovali, L.Ricci *DIVES: A Distributed Support for Networked Virtual Environments* 20th IEEE AINA 2006, Wien, April 2006

[6] J. Keller, G. Simon *Toward a Peer-to-Peer Shared Virtual Reality* 22nd International Conference on Distributed Computing, July 2002

[7] J. Liebeherr, M. Nahas *Application LAyer Multicast with Delaunay Triangulations* IEEE Journal on Selected Areas in Communications 40(8), October 2002

[8] E. Kranakis, H. Singh, J.Urrutia *Compass Routing on Geometric Networks* 11th Can. Conf. on Computational Geometry, CCCG 99, August 1999

[9] F. Aurenhammer *Voronoi Diagrams-A Survey of a Fundamental Geometric Data Structure*. ACM Computing Surveys, Vol. 23, n. 3, Sept. 1991.

[10] S.Hu, J.Chen, T.Chen *VON: A scalable peer-to-peer network for virtual environments*, IEEE Network, vol. 20, n. 4, Jul/Aug. 2006.

[11] A. Yu, S. T. Vuong *MOPAR: a mobile peer-to-peer overlay architecture for interest management of massively multiplayer online games*, NOSSDAV 05 Washington, June 2005, pp. 99-104 .

[12] J. Jiang, Y. Huang, S.Hu, *Scalable AOI-Cast for Peer-to-Peer Networked Virtual Environments* 28th ICDCSW, June 2008.

[13] J. Chen, W. Lin, T. Chen, S. Hu, *A Forwarding Model for Voronoi-based Overlay Network* 13th ICPADS 2007, P2P-NVE work., December 2007.

[14] J.Lee et al. *APOLO:Ad hoc Peer to Peer Overlay Network for Massively Multi-player Online Games*, Technical Report, KAIST, 2006

[15] The Peersim Simulator http://peersim.sourceforge.net/