

# AOI cast by Tolerance Based Compass Routing in Distributed Virtual Environments

Michele Albano and Antonio Quartulli and Laura Ricci  
 Dip. di Informatica, Università degli Studi di Pisa, Italy  
 {michele,ricci}@di.unipi.it, antonio.quartulli@studenti.unitn.it

Luca Genovali  
 IMT, Institutions, Markets, Technologies, Lucca, Italy  
 l.genovali@imtlucca.it

**Abstract**—This paper presents an Area of Interest(AOI)-cast strategy for P2P Distributed Virtual Environment (DVE) which exploits a Delaunay Triangulation of the DVE to define a compass-based routing algorithm. A set of formal results for circular AOI is presented. Inconsistencies between local views of different peers due to the network latency are faced by introducing a tolerance threshold in the compass routing.

## I. COMPASS ROUTING IN VORONOI DVES

A *Voronoi tessellation* [3] partitions a *DVE* into  $k$  areas such that the area corresponding to a peer  $p$  includes all the points of the *DVE* which are closer to  $p$  than to any other peer. Two peers are *Voronoi neighbours* iff the borders of their areas overlap. The connected graph defined by linking neighbouring peers is the *Delaunay Triangulation* corresponding to the *Voronoi tessellation*. Recently, several Voronoi based DVE [1], [5] have been proposed.

*Compass Routing* [2] is a routing algorithm for geometric graphs based on the following observation. Consider a connected graph and assume of being located at one of its nodes  $n$  with the goal to reach a destination node  $d$ . [2] shows that the best strategy is to look at the edges incident in  $n$  and choose the edge whose slope is minimal with respect to the segment connecting  $n$  and the destination  $d$ . [2] 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*. [3] exploits compass routing to define a *Spanning Tree* supporting an *application level multicast*.

We propose to exploit compass routing to define an *AOI-cast mechanism* for heartbeats propagation. The root of the multicast tree is the peer which generates the heartbeat and the tree includes the peers belonging to its *AOI*. We describe our algorithm through the example shown in Fig.1. Let us suppose that the peer *Root* generates an heartbeat, i.e. it is the root of the spanning tree, and let us consider the peer *A* which receives the heartbeat. *A* should choose among its Voronoi neighbours its children in the spanning tree. For instance, to decide if peer  $A_5$  is its child, the algorithm evaluates whether  $A_5$  would have chosen *A* in its path toward the root *Root*. This would happen if  $\angle RootA_5A$  is smaller than  $\angle RootA_5A_1$  and  $\angle RootA_5A_4$ . Note that if this happens,  $A_5$  does not need to compare the slopes of the edges connecting  $A_5$  to further neighbours with respect to the segment  $\overline{RootA_5}$ , since they are surely larger. This argument is reversed in order to detect the children of a peer in the spanning tree. The basic point

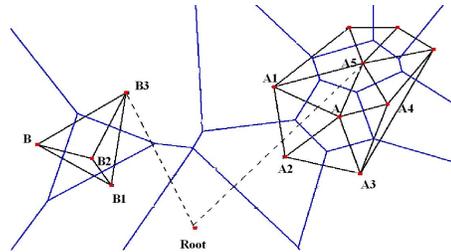


Fig. 1. Spanning Tree Construction

is that a peer  $p$  detecting whether its Voronoi neighbour  $v$  is its child in the spanning tree must determine the vertexes of the Delaunay Triangles sharing the Delaunay edge  $\overline{pv}$  before applying the angle evaluation phase. For instance, in Fig. 1 *A* should consider the triangles  $AA_5A_1$  and  $AA_5A_4$  to detect whether  $A_5$  is its child. Note also that *B* should consider only the triangle  $BB_2B_3$  to detect whether  $B_3$  is its child in the tree, because of the borders of the *DVE*. Notice that it is not possible to identify the Delaunay triangles to be considered from the Voronoi neighbourhood relation alone. For instance  $B_1$  and  $B_3$  in Fig. 1 are both Voronoi neighbours of *B* and Voronoi neighbour themselves, but the triangle  $B_1, B, B_3$  does not belong to the Delaunay triangulation. A detailed description of the algorithm appears in [4] where a set of simple conditions to check if two consecutive neighbours of a node  $n$  define a Delaunay triangle together with it is given.

## II. COMPASS ROUTING IN CONSTRAINED AREAS

The spanning tree supporting the *AOI-cast* should include the nodes corresponding to the peers located in the *AOI* of the root peer. This implies that compass routing may require to step out the *AOI* in order to build a spanning tree covering all its peers. It is interesting to evaluate the number of external peers which should be visited, because each of these links implies performing a routing hop so introducing a further latency in the delivery of an heartbeat.

Let  $D$  be a *DVE* including a set of nodes. If we consider a subregion  $A$  of  $D$  we can define  $G(A)$  as the graph comprising the nodes of the  $D$  that belong to  $A$  and the subset of Delaunay links whose end points both belong to  $A$ . Note that, in the general case,  $G(A)$  may be not connected or it may not be a Delaunay triangulation, because the convex hull of the nodes

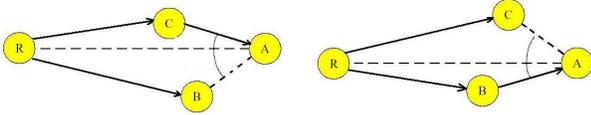


Fig. 2. Local View Inconsistencies: view of B(left) and view of C(right)

may include links not belonging to  $G(A)$ . For these reason, when compass routing is exploited to define a spanning tree rooted at one of the nodes in  $A$  and covering all the nodes in  $A$ , it should consider, in the general case, a set of nodes and links not belonging to  $G(A)$ .

The results in [4] show that compass routing is able to build a spanning tree including all and only the nodes of the  $AOI$  when the  $AOI$  has a *circular shape*. Hence, no hop outside the  $AOI$  is required in this case and latency is not increased. The first result shows that the graph  $G(A)$  is always connected when  $A$  is a circular region.

*Theorem 1:* Let  $D(R)$  be a Delaunay triangulation defined on a set of nodes belonging to the 2-dimensional space  $R$ . If  $A$  is circular shaped subregion of  $R$ , then  $G(A)$  is connected.

Even if the  $G(A)$  is connected, compass routing may require to consider some Delaunay links not belonging to  $G(A)$  to build the spanning tree. The following theorem shows that this is not necessary if a circular region is considered.

*Theorem 2:* Let  $D(R)$  be a *Delaunay Triangulation* defined by a set  $S$  of sites belonging to a 2D space  $R$ . If  $A$  is a circular shaped subregion of  $R$  centered on the node  $s \in S$ , compass routing is able to compute a spanning tree rooted at  $s$  and including *all and only the sites* of  $S \in A$ .

These results guarantee that the algorithm described in Sect.1 does not require to step out the  $AOI$  to build the spanning tree. Furthermore, the last theorem suggests that any peer belonging to the  $AOI$  of a peer  $P$  should consider, in the angle evaluation phase of the spanning tree construction, its *Voronoi neighbours* belonging to the  $AOI$  of  $P$  only. As a matter of fact, peers located outside the  $AOI$  cannot belong to the spanning tree and should not be considered. The theorems are not valid for areas of interest of different shape. For instance, if rectangular or squared areas are considered, some paths of the spanning tree may zig zag in and out the borders of the considered region.

### III. TOLERANCE BASED COMPASS ROUTING

The definition of a routing algorithm for  $DVEs$  must take into account the inconsistencies which may arise because of the movement of the peers. As a matter of fact two peers may have a different perception of the position of a common neighbour, due to the delay of the notifications. This implies that these peers may perceive a *spatial drift* with respect to the real position of their common neighbour. Consider, for instance, the scenarios shown in Fig. 2 where the left one

corresponds to the local view of the peer  $B$  while the right one to that of  $C$ . If  $B$  and  $C$  receive an heartbeat from their common neighbour  $R$ , they both neglect to propagate the heartbeat to  $A$  because of their different views of the  $DVE$ . Due to the *spatial drift*, each peer supposes that the other one should propagate the heartbeat to  $R$ . As a matter of fact, compass routing at peer  $B$  decides to neglect the propagation of the heartbeat because the slope of the segment  $\overline{AC}$  with respect to the segment  $\overline{AR}$  is smaller than that of the segment  $\overline{AB}$ , the other way around for peer  $C$ . The *spatial drift* may also generate *redundant notifications*, because  $B$  and  $C$  may decide to propagate the same heartbeat to their common neighbour. This scenario occurs when the local views in Fig.2 are inverted. Note that these problems are introduced by the highly dynamic nature of the  $DVE$ . It is worth noticing that the first problem is more serious, since it may lead to the overlay partition. We have modified the algorithm described in Sect.I to reduce the number of the peers which do not receive an heartbeat. Our strategy is to define a constant networkwide *tolerance threshold*: a peer states that one of its Voronoi neighbours is its child in the spanning tree whenever the difference between the angles considered by compass routing is lower than a *tolerance threshold*. Note that in this case an heartbeat may be notified to a peer by more than one neighbour. For instance, in Fig.2, both  $B$  and  $C$  should send the heartbeat to  $A$  if the difference between the angles is lower than the threshold. As a consequence, the resulting algorithm introduces a number of redundant messages. Anyway, in our case, it is better to send a larger number of messages, instead of having some peers that do not receive the heartbeat at all.

A further mechanism is introduced to reduce the probability of partitioning the overlay when the  $DVE$  is scarcely populated, i.e. the opposite scenario w.r.t. to crowding. In this case the  $AOI$  of a peer  $P$  may be empty but  $P$  sends its heartbeats to its Voronoi neighbours anyway so that the topology of the Delaunay overlay is dynamically maintained. However, overlay disconnections may still arise because of the spatial drifts or of unexpected peers crashes. To reduce the probability of this scenario, a  $TTL$  is paired with each heartbeat and the algorithm is modified such that a heartbeat propagates on as long as the border of the  $AOI$  is not reached or the value of the  $TTL$  is  $\neq 0$ . If the  $AOI$  of a peer is crowded, the heartbeat is propagated only inside it, otherwise if it is scarcely populated, it is propagated at least  $TTL$  hops away from its source.

### REFERENCES

- [1] R. Cavagna, M. Abdallah, E. Buyukkaya, and C. Bouville. Vorogame: A hybrid p2p architecture for massively multiplayer games. In *IEEE CCNC*, January 2009.
- [2] E.Kranakis, H.Singh, and J.Urrutia. Compass routing on geometric networks. In *11th Can. Conf. on Computational Geometry, CCCG*, August 1999.
- [3] F.Aurenhammer. Voronoi diagrams-a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23, September 1991.
- [4] L. Genovali. *A Voronoi Based Framework for the Definition of P2P Distributed Virtual Environments*. PhD Thesis, IMT, July 2009.
- [5] S.-Y. Hu, J.-F. Chen, and T.-H. Chen. Von: A scalable peer-to-peer network for virtual environments. *IEEE Network*, July-August 2006.