1. EXTENDED ABSTRACT

Distributed Virtual Environments (DVE) such as military or civil protection distributed simulations and massively multiplayer online games (MMOG), for instance World of Warcraft or Second Life, are currently gaining increasingly attention in the software market.

Currently the client server model is still the reference computational model for this kind of applications, but approaches based on the P2P model [4, 2, 1, 3] have recently been proposed. The definition of a fully distributed architecture for DVE is an actual challenge because of the complexity of these applications which integrate networks, graphics and AI programming. On the other hand, the adoption of a distributed computational model is mandatory to overcome the low scalability of client server architectures.

In a DVE, a set of active entities, represented by avatars, interact with each other and with a set of passive objects, like weapons, potions, etc. Each avatar moves continuously within the DVE and in general it interacts with the avatars and the objects located in its surroundings only. This locality property which is modeled through the notion of Area of Interest (AOI) of an avatar, should be properly exploited by a communication support to reduce the amount of messages exchanged through the P2P overlay. On the other hand, since the view of each avatar is constrained by its AOI, a mechanism to dynamically acquire knowledge of avatars and of passive objects beyond the AOI is required.

The management of the state of the passive objects of the DVE is a further critical issue for the development of P2P supports because it requires to manage the concurrent updates that may occur when several avatars concurrently update the same object. The adoption of classical solutions like those based on logical timestamps is not suitable in this case, due to the high number of messages required to implement these mechanisms. Furthermore, since P2P networks are highly dynamic, a set of mechanisms to guarantee object persistence should be defined as well.

We are currently investigating the feasibility of exploiting Voronoi Tessellations [5] to model DVE applications. Given a set of sites \( S = s_1, \ldots, s_n \) in a 2D space, a Voronoi tessellation is a space partition that assigns to each site \( s_i \) the region \( \text{Voro}(s_i) \), that includes the set of points which are closer to \( s_i \) than to any other site \( s_j \in S, i \neq j \), according to a given metric. Standard euclidean metric is exploited in classical Voronoi tessellations. Two sites are Voronoi neighbours iff the borders of their region overlaps. A Delaunay triangulation is a graph that connects the Voronoi neighbors. Fig. 1 shows a 2D Voronoi Tessellation.

We propose to model a DVE by a Voronoi Tessellation where sites correspond to peers and the Delaunay triangulation links define the topology of the P2P overlay. Any passive object in \( \text{Voro}(s_i) \) is assigned to the peer corresponding to \( s_i \), which becomes the object owner, stores its state and manages concurrent updates.

In a DVE, any peer \( P \) periodically sends a heartbeat, i.e. a message notifying its position to any other peer in its AOI. We have exploited compass routing, [6] an algorithm that properly exploits the property of
**De}alnay triangulations to compute a multicast spanning tree covering the peers belonging to \( AOI(P) \). A large number of hops may be required to reach any peer in \( AOI(P) \) especially when a crowding scenario occurs, i.e. a huge amount of peers is located in \( AOI(P) \). Note that crowding often occurs in a DVE, where peers may either be attracted by the same object, for instance a sword or a magic potion, or gather to fight each other. Since a large delay may have a negative impact on the responsiveness of the application, any peer may dynamically define direct connections with a subset of peers in its \( AOI \). These connections are added to the Delaunay links and act as 'shortcuts' by reducing the delay.

The most challenging issue of our approach is the definition of proper distributed algorithms to guarantee that the structure of the Delaunay overlay is correctly preserved and no overlay partition occurs. We propose a 'pass the word' approach, where peers becomes acquainted with each other through their Voronoi neighbours. A peer receiving a heartbeat from one of its neighbours \( N \) checks if any of its further neighbours \( Q \), \( Q \neq N \), is entering \( AOI(N) \), and in this case it propagates the heartbeat generated by \( N \) to \( Q \). In this way, each peer acts as a beacon for its neighbours by 'putting in touch' peers that are not acquainted with each other. A similar approach is adopted to acquire new objects located beyond the \( AOI \) of a peer.

It is worth noticing that several inconsistencies may rise due to network delays, messages loss, or abrupt peer crashes which often occurs in P2P environments. A certain amount of replication is required to avoid irrecoverable situations due to these inconsistencies. For instance, it is necessary to guarantee that each peer is always connected to its Voronoi neighbours to avoid the partitioning of the overlay. This may be obtained, for instance, by sending redundant heartbeat messages of a peer \( P \) to the Voronoi neighbour of its neighbour, even when the latter do not belong to its \( AOI \).

We are currently investigating the feasibility of exploiting Weighted Voronoi Tessellations [5] to model hierarchical P2P networks. A Weighted Tessellation assigns a weight \( w_i \) to each site \( s_i \), so that the size of \( Voro(s_i) \) depends upon \( w_i \). The metric exploited by Additively Weighted Voronoi Tessellations, AWV, defines the distance of a point \( P \) from a site \( s_i \) as the sum of \( w_i \) and the euclidean distance between \( P \) and \( s_i \). Fig. 2 shows an Additively Weighted Tessellation where the weight of a peer \( P \) is proportional to the extension of the circle centered at \( P \). Notice the presence of peers which have been 'absorbed' by peers with a larger weight located in their surroundings. No Voronoi region is associated with these peers which are hidden to the rest of the world and do not belong to the Triangulation.

We exploit AWV tessellations to model a hierarchical P2P Network, where the weight assigned to each peer is proportional to its computational power. The adoption of AWV Tessellations to model hierarchical P2P overlays offers several advantages. First of all, larger Voronoi regions are assigned to peers characterized by larger weights. This defines a load balancing strategy for passive objects, because the number of passive objects assigned to each peer will be proportional to its weight. Since no area is assigned to hidden peers, they will not manage any object. Furthermore, peers which have 'absorbed' other peers become Superpeers acting as servers toward them. An hidden peer \( P \) characterized by a low weight may rely on the corresponding Superpeer which acts as a proxy for \( P \) by forwarding any notification generated by \( P \) to its neighbours and by notifying to \( P \) events generated in its surroundings. Note that the same peer may play a different role according to the DVE scenario. For instance, the same peer may act as a superpeer if the number of neighbours is bounded by a threshold, while it requires the support of a superpeer if this number exceeds this threshold.

2. REFERENCES