Toward Community-Driven Interest Management for Distributed Virtual Environment*

Emanuele Carlini[1], Patrizio Dazzi[1], Matteo Mordacchini[2], and Laura Ricci[3]

[1] ISTI – CNR, Pisa, Italy
[2] IIT – CNR, Pisa, Italy
[3] University of Pisa, Italy

Abstract. A fundamental requirement for the discovery and the retrieval of the entities of a distributed virtual environment is the definition of a scalable *Interest Management (IM)* support. This paper presents a distributed protocol for IM that integrates two different gossip protocols. The first one based on a discrete tessellation of the Area of Interest of a peer in order to maximise its coverage by the peer neighbours. The second is based on a distributed algorithm enabling the peers to self-organise themselves in communities. The paper presents a set of experimental results showing the effectiveness of the proposed approach.

1 Introduction

Collaborative Virtual Environments (CVEs) are softwares for supporting the collaboration of many participants located worldwide. Typical examples of CVEs are distributed computer simulations, multi-player on-line games, and collaborative engineering software. The implementation of a CVE exposes a set of entities (which represents the virtual world) and that can be accessed by the participants via network.The virtual representation of a participant in a CVE is called *Avatar*, a virtual alter-ego enabling the interaction with other avatars (participants) and with the passive entities in the CVE. Since the early nineties, a lot of solutions have been proposed to give a distributed organisation to virtual environments. Because of the communication latencies (and the limitations on available network bandwidth, especially in the case of asymmetric network connections, e.g., ADSL), careful attention should be paid for implementing proper data management, consistency and transfer models. To reduce the impact of network related limitations the information about the entities is replicated in several machines. Hence, the general issue is the *discovery* and *retrieval* of the entities in a scalable fashion. This issue is usually referred as Interest Management (IM), and it can be abstracted using a publish-subscribe model [1]. Publishers perform actions (e.g., they move) and subscribers should receive the information about these actions. IM poses well known scalability challenges, in particular when number of participants grows. A large numbers

* This work is funded by the EC under the CONTRAIL (FP7-257438), RECOGNITION (FP7-IST 257756) and EINS (FP7-FIRE 288021) projects.

D. an Mey et al. (Eds.): Euro-Par 2013 Workshops, LNCS 8374, pp. 363–373, 2014.

of approaches have been proposed to efficiently support IM (see [2, 3] for a comprehensive list). One of the main strategies to address IM is to replicate only the entities in the visual/interaction area of players, which is called Area of Interest (AOI) that, typically, consists of a circular area whose centre is the avatar position. In this paper we propose a community based IM in which avatars self-organise themselves in communities by exploiting gossip based protocols. In particular, we limit the IM to the positions of the avatars. Anyhow, we believe that our approach is general enough to be extended and applied to other types of information. Our approach employs an eventually consistent model for the positions of the avatars. According to this model, an avatar changes the value associated to its own position in its local storage contextually with its actual movement. Later, interested peers are informed about the changed avatar position. Gossip protocols are a class of unstructured peer-to-peer communication protocols which involve periodic message exchanges between pairs of peers, eventually resulting in information being spread throughout the system in a way that is pretty similar to human gossiping. The gossip protocols are gaining greater and greater popularity in distributed systems scenarios. Indeed, they demonstrated to be highly efficient solutions for information dissemination and aggregation [4–7]. Moreover, they allow a high degree of scalability and reliability even in constantly changing environments. This is why we decided to adopt them in our proposal for a peer-to-peer IM addressing DVEs. In a previous work of ours [8] we sketched the *Coverage Peer Sampling* protocol: our first attempt regarding the adoption of the gossip protocols to DVEs. The key idea consisted in the tessellation of the AOI of each peer and the organisation of the peer neighbourhood so that the coverage of the peer AOI by its neighbours is maximised. Although such approach showed up encouraging results, it leads peers in adopting a selfish approach in building their neighbourhood. Even if this is not a big issue in several DVE related scenarios, sometimes it may lead to an inefficient percolation of the information flow in the network. In this paper we adjust such approach equipping it with the ability of exploiting network-wide information for organising the peers' neighbourhood that enhance the overall network structure. To this end, we added to each node belonging to the DVE a gossip protocol for network-wide clustering of peers in communities on the basis of their spatial location. The information about the communities are then exploited by the Coverage Peer Sampling protocol to optimise its neighbourhood. This integration is not particularly disruptive since the "stack-ability" property of the gossip protocols, namely the possibility to organise them in stacks that share a common information space but with each layer devoted to a specific goal. This property of Gossip protocols has been exploited in the past for realising quite complex solutions such as ring topologies [9] as well as community building protocols [10, 11]. As community building protocol we chose GROUP [11], a gossip-based approach for distributed community construction that through a voting phase identifies a set of "leaders" which become the representatives of the community. The rest of the paper is organised as follows. In Section 2 we show alternative solutions for IM in DVE and for community recognition in P2P. Section 3 describes the overall architecture of the proposed protocol. A preliminary performance evaluation is presented in Section 4. Finally, in Section 5 we draw our conclusions.

2 Related Work

Discovery and retrieval of resources, entities and/or services is a very broad topic in the P2P literature (e.g. [12–14]). Specifically, IM solutions proposed in the literature for P2P-based DVE can be classified according to the different roles played by the nodes of the infrastructure. Existing solutions span from hierarchical approaches, in which IM is realised through a set of supernodes, to flat approaches in which there is no clear distinction between super and regular nodes. The solutions based on supernodes (e.g., [15, 16]) exploit user-provided resources to realise IM. The virtual environment is partitioned into regions, each one paired with a supernode that periodically informs the peers about their neighbours. In order to cope with the unreliability, these approaches needs to carefully select the supernodes, moreover, failure recovery mechanisms and adaptive mechanisms for region sizing are also needed. For their intrinsic nature, flat approaches pairs with DVEs that are purely P2P. Flat approaches can be either structured or unstructured. Colyseus [17] uses a DHT to store the state of the entities. DHTs offer concrete advantages, e.g., a stable and reliable platform for distributed indexing. However, their latencies may represent an issue. Compared to DHTs, unstructured overlays (such as [15, 18]) give more emphasis to the dynamic grouping of peers. Many of these approaches exploits direct connections among peers that are close one anothers. This strategy pays a reduced network latency, since each recipient is often just one hop away from the source. Other approaches [19, 20] exploit the position of the avatars to define a Voronoi partition of the virtual world to define the overlay connections. As we stated in the introduction, in this paper we combine the Coverage Peer Sampling approach for maximising the AOI coverage with GROUP. GROUP demonstrated to be an efficient solution for community building, anyway in the scientific literature there can be found other approaches. In the following of this section we report two notable solutions that have been proposed in the past. *Ramaswamy et al.* [21] propose CDC, an approach for clustering peers. Like GROUP it assumes that each node only knows its direct neighbours. CDC achieves a network-wide clustering from a set of starting nodes, that behave as aggregators. Each aggregator initiates a stream of votes. The protocol ends up with peers choosing as their own cluster the one associated with the neighbour that received the highest amount of votes. A critical aspect of this solution is that the division of the nodes in clusters is highly dependent on the choice of the aggregators. As a consequence, with this kind of approach it may happen that the initiators nodes are different from the ideal ones to represent the cluster. *Datta et al.* [22] propose LSP2P and USP2P, two approximations of K-Means for P2P environments. LSP2P is based on a gossip protocol to distribute the centroids. The main limitation of this protocol is the assumption that data is uniformly distributed among the peers, i.e., each peer has at least some entry from each cluster. This assumption not necessarily hold in our scenario. The second algorithm, USP2P, uses sampling to give probabilistic guarantees. However, these guarantees are also based on the assumption that data is uniformly distributed among peers. In addition, USP2P requires a set of coordinating peers tath get easily overloaded, since they are responsible for exchanging centroids with a significant number of peers for sampling.

3 Protocol Architecture

Figure 1 shows the stack of gossip-based protocols that every peer in DVE executes. On the bottom we have a random peer sampling (RPS) layer, providing a random view of the system to the other layers. RPS is useful to maintain connectivity, and to prevent the other layers from getting stuck in local solutions. The semantic-driven peer sampling exploits the RPS to build a view of the system which is based on a semantic distance. This is in turn used by the GROUP protocol to elect leaders and build communities. Section 3.2 provides more details about the GROUP protocol. The last component of the architecture is the coverage peer sampling (CPS) layer. The base CPS considers the information coming from the RPS and also executes a gossip protocol to provide the peer with a view of peers that potentially contains relevant entities for its AOI. The enhanced version of the CPS, which we call CPS+, considers also the global information coming from the GROUP layer to compute the view of peers in the DVE. More details on the CPS and CPS are provided in Section 3.3.

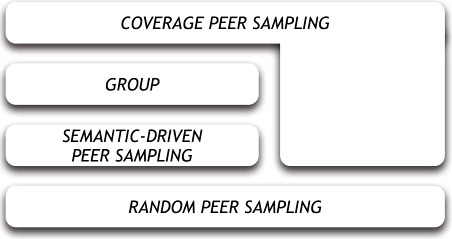

Fig. 1. Gossip Layers

3.1 Gossip Protocols

Gossip protocols are a set of very effective, flexible, yet simple tools for exchanging information and for overlays creation and management purposes. The simplicity of gossip protocols relies on the fact that each node acts on a pure local information basis. The effectiveness of gossip is highlighted by the fact that, using only local interactions, nodes are able to achieve global results, like the maintenance of a connected network. Gossip protocols offer the advantage of reducing the number of messages required to spread information in a P2P network if compared with pure unstructured protocols. Even though their associated communication overhead could be higher than that of structured protocols,

```
while true do
    sleep T seconds;
    $q \leftarrow$ selectPeer();
    $sendBuffer \leftarrow$ ItemsToSend();
    send sendBuffer to $q$;
    receive receiveBuffer from $q$;
    $view \leftarrow$ ItemsToKeep();
```

Fig. 2. Active Thread

```
foreach Message m Received do
    extract sender $p$ from $m$;
    $sendBuffer \leftarrow$ ItemsToSend();
    send sendBuffer to $p$;
    $view \leftarrow$ ItemsToKeep();
```

Fig. 3. Passive Thread

gossip-based approaches are more efficient in dealing with networks composed of highly dynamic nodes. When exploiting a gossip-based protocol, each node in a P2P network maintains only a limited, partial, *view* of the network, i.e., a small set of *neighbour* nodes. It periodically exchanges information with another peer properly chosen among its neighbours. This simple, repeated, local data exchange allows the whole network to achieve specific goals and results. Each gossip protocol can be designed using the two skeletons depicted in Figures 2 and 3. Since the functions selectPeer(), ItemsToSend() and ItemsToKeep() determine the peers to communicate with as well as the information exchanged and kept by peers, by changing these methods it is possible to achieve different goals or different overlay organisations. As we already mentioned, the gossip protocols can be composed into layered architectures, in which each layer is devoted to a specific aim. According to this view, each gossip layer has two main duties:

- to maintain and to exchange information regarding the task it is devoted to;
- to feed with proper information the higher gossip levels, if any;

Examples of gossip-based protocols are randomised peer sampling services [23, 24], used to maintain connected networks, even in face of highly dynamic nodes. Another relevant example is given by Vicinity [25], a similarity-based node aggregation protocol, built on top of Cyclon [24].

3.2 GROUP

The local knowledge maintained by each peer typically does not allow a good identification of the features which may characterise communities of peers. In fact, there is not any explicit identification and characterisation of possibly larger groups of peers sharing a certain degree of similarity, such that they can be considered as a community of affined entities. In order to obtain such an explicit community recognition, we want to employ a protocol that exploits the collaboration among peers in order to reach a collaborative definition of the communities and their representation. GROUP is a P2P gossip protocol build on top of a similarity gossip-based set of layers, whose role is to build communities of similar peers and give a way to characterise them. This goal is achieved through the identification of a set of communities representatives that peers select using a P2P voting phase driven by the consensus a peer profile can gather among the other nodes. To this end, each peer gives a bunch of votes to the nodes having a profiles that, at best of its knowledge, are the most similar to its own. The most voted peers are then elected as representatives and, together with the peers that contributed to their election, constitutes a community. A community is simply identified by the profile of its representative peer. The detection of the representative of each community is structured in three following stages: (i) Similar Peers Detection, (ii) Potential Candidates Selection, (iii) Representative Election.

(i) *Similar Peer Detection*: In this phase, the active thread of each peer selects and informs the k most similar peers in its neighbourhood that it is considering them as the most similar ones among its neighbours.

(ii) *Potential Candidates Selection*: The aim of the second phase is to exploit the information coming from the previous one in order to detect the potential community representative of each peer. In this step, each peer selects from its neighbourhood a potential candidate to represent itself, hence to be the representative of its community. Each peer chooses such a potential candidate among the neighbours that have received in the previous phase a number of votes higher than a certain threshold.

(iii) *Representative Election Phase*: In this last phase each peer p chooses as its representative the node R that has collected the highest number of *"potential candidate"* votes. If there are two or more peers that received the same number of votes, p chooses the one with the most similar profile to its own.

It is worth to point out that the members of the communities built by GROUP are not required to keep track of the community structure. No peer is required to have an explicit knowledge about all the other peers in the community. The representative itself is able to recognise (and elect) itself as such by counting the votes it had received, without any other needed interaction and knowledge. Peers label themselves with the community profile in order to recognise themselves as members of the same group.

3.3 Vitruvian

The goal of this layer is to obtain a selection of peers that maximise the retrieval of entities in the virtual environment. In order to better explain this task, let us consider this problem from the point of view of a generic avatar A. At an arbitrary point in time, A has in its local representation of the virtual environment the replicas of the entities that belong to its AOI. When A moves across the virtual environment, its AOI changes accordingly. Hence, to maintain its local representation up-to-date, the peer (P) represented by A must discover the new entities belonging to the new AOI. In order to dynamically acquire these new entities, P must discover the peers that can efficiently provide such information. We call this approach *Coverage Peer Sampling* (CPS). The CPS consists in the implementation of the three functions that characterise any gossip-based sampling protocol (as described in section 3.1), namely $selectPeer()$, $ItemsToSend()$, and $ItemsToKeep()$. In this context, the criterion driving the CPS is the *area coverage*, which is defined as the following. Given a set S of peers and a pivot peer P, the area coverage can be defined as the intersection between the union of the AOIs of the peers in S and the AOI of P. Intuitively, a peer would maintain such a view that maximises the coverage of its own AOI, so to have higher chances to obtain relevant information about entities in its proximity.

selectPeer() This function chooses from the view of P the target peer for the gossip communication, which we refer to as T in the following. If there are peers in the view of P that are contained in its AOI, the algorithm selects the one that maximises the euclidean distance with the current position of P. Conversely, the algorithm selects the peer that minimises the euclidean distance with the current position of P. The rationale behind this behaviour is that, if P has several peers in its AOI, it probably has some knowledge of its immediate proximity. Choosing a peer at the borders of its AOI increases the knowledge of a region that could be explored in the next close future. On the other hand, if no peers are in the AOI of P, a closer peer might have information about a region that P is going to visit soon.

ItemsToSend() Once T has been selected, P has to select the "best" subset of peers in its view to send to T. P ranks the peers in its view by evaluating their AOIs with respect to the AOI of T. To find the optimal subset of peers is a NP-hard problem[1]. Since this function has to be repeated multiple time in a second, the *ItemsToSend()* is implemented as an heuristic, which partitions the AOI's areas into a set of discrete *tiles*. The heuristic takes as input the AOI of T and all the AOI of the peers that compose the view of P, then assigns a score to each tile which is the reciprocal of the number of intersected AOIs. Once the tiles have been assigned with a score, the heuristic computes the score of a peer as the sum of the scores of each intersected tile. Then, the peers are sorted according to their score and inserted in a list L. In this point of the heuristic lies the difference between CPS and CPS+. CPS just returns the first d items of L (where d is the size of the subset to choose). In other words, CPS considers only the area coverage as the criteria for the selection of the peers. Instead, CPS+ also considers the global knowledge of communities, which is provided by the GROUP layer, for the selection of the subset. CPS+ chooses from L the d peers with the highest score that maximises the number of unique communities. This implies that peers with a high score can be discarded in favour of a peer with lower score but with a valuable knowledge about a community.

ItemsTokeep() From a logical point of view, this function shares the same principles of the $ItemsToSend()$ one. The only difference is the input: (i) P selects itself as pivot, and (ii) the set of peers to rank includes the peers coming from the random peer sampling layer.

4 Experimental Results

To simulate a region of a VE we consider a squared area of 1200 x 1200 points. The region has 10 circular fixed hotspots, which represents popular places inside the VE. Avatars have an AOI of 50 points, and move on the map according to realistic mobility traces that have been computed according the mobility model

[1] It is easy to demonstrate the equivalence with the *maximum coverage problem*.

presented by Legtchenko et al. [26], which simulates avatars movement in a commercial MMOG[2]. The model works according to the hotspots defined in the region. When an avatar reaches a hotspot, it explores the hotspot for a span of time and eventually it moves to another hotspot. In the simulation all the gossip layers share the same configuration. The cache of the peer is the square root of the total number of peers. The simulation is divided into tick, each tick being 500 milliseconds. The simulations ran on a machine equipped with Java 7, 16GB of RAM, an Intel i5-2550 quad core @3,30 Ghz. To simulate the gossip communications we used Epeerdemics [27]. Figures 4 and 5 show the probability density function (PDF) of the average area coverage of respectively CPS+ and CPS. Note that the CPS+ approach achieves better results. More in detail, it can be observed that the peers reaching an area coverage around 96% in CPS are about 20%, whereas with CPS+ are about the 27%.

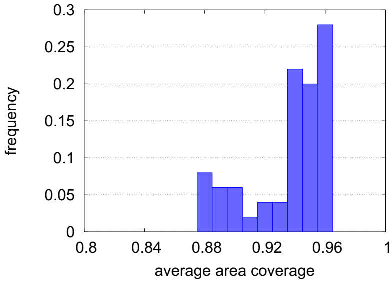

Fig. 4. CPS+

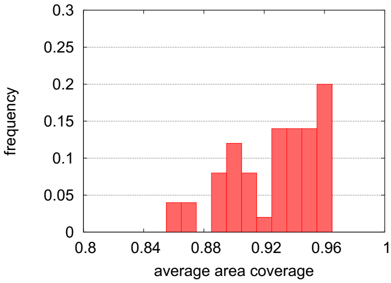

Fig. 5. CPS

More generally, the values of the area coverage are distributed more toward the right side of the spectrum with CPS+ rather than with CPS, highlighting the ability of the proposed system to allow the nodes in the network to achieve a better awareness of their surrounding environment. The second set of experiments compare the CPS and the CPS+ in terms of the overlay they build. We model the overlay as a graph, which is constructed by considering the peers as vertexes and a directed edge between two peers A and B exits if A has B in its view or viceversa. Clearly, a different graph exists for every iteration of the protocol. Our analysis compares the two approaches by analysing the above features in 50 iterations.

Figure 6 shows the average clustering coefficient (CC). Besides an initial startup time, the value of the CC remains stable on the same value in both cases. However, in this case CPS+ shows more variability. It is interesting to notice how the value of CPS+ decreases before the 20th and 40th iterations and then it grows up again. In fact during these iteration, the GROUP protocol elects the new leader. Figure 7 shows the evolution of the diameter of the overlay in the first iteration. After the start-up period, the diameter remains fairly stable after the 10th iteration. In particular CPS has a diameter of 4 while the CPS+ of 3.

[2] Second Life, `http://secondlife.com/`

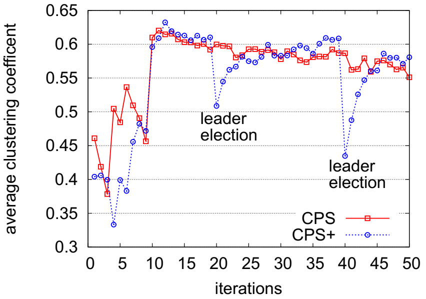

Fig. 6. Clustering coefficent

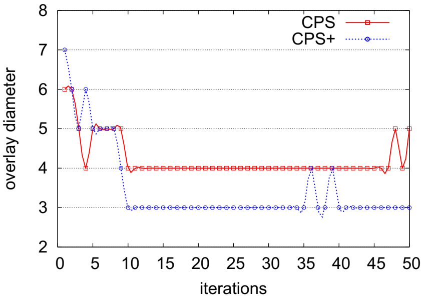

Fig. 7. Diameter

5 Conclusions

In this paper we propose a Interest Management solution for large scale DVEs based on self-emerging communities of avatars. The proposal regard the avatars in a DVE as nodes in a P2P network and exploits a gossip-based interaction among avatars in order to let them self-organise in communities of peers that are exploring nearby locations. Preliminary experimental results show that the proposed approach is able to reduce the diameter of the network graph, improving the results of the previously proposed gossip-based solution that does not create communities. At the same time, it is able to guarantee to each peer a better coverage of the its AoI. Starting from these results, we plan to further analyse this solution by considering larger network sizes. Moreover we plan to study the behaviour of the system in more challenging conditions, like a heavy churning nodes scenario and we are trying to give a formal mathematical definition of the community creation gossip protocol.

References

1. Hu, S.: Spatial publish subscribe. In: Proc. of IEEE Virtual Reality (IEEE VR) Workshop, Massively Multiuser Virtual Environment, MMVE 2009 (2009)
2. Ahmed, D., Shirmohammadi, S.: A dynamic area of interest management and collaboration model for p2p mmogs. In: Proceedings of the 2008 12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications, pp. 27–34. IEEE Computer Society (2008)
3. Boulanger, J., Kienzle, J., Verbrugge, C.: Comparing interest management algorithms for massively multiplayer games. In: Proceedings of 5th ACM SIGCOMM Workshop on Network and System Support for Games, p. 6. ACM (2006)
4. Jelasity, M., Montresor, A., Babaoglu, O.: Gossip-based aggregation in large dynamic networks. ACM Transactions on Computer Systems (TOCS) 23(3), 219–252 (2005)
5. Dazzi, P., Felber, P., Leonini, L., Mordacchini, M., Perego, R., Rajman, M., Rivière, É.: Peer-to-peer clustering of web-browsing users. In: Proc. LSDS-IR, pp. 71–78 (2009)

6. Mordacchini, M., Dazzi, P., Tolomei, G., Baraglia, R., Silvestri, F., Orlando, S.: Challenges in designing an interest-based distributed aggregation of users in p2p systems. In: International Conference on Ultra Modern Telecommunications & Workshops, ICUMT 2009, pp. 1–8. IEEE (2009)
7. Baraglia, R., Mordacchini, M., Dazzi, P., Ricci, L.: A p2p recommender system based on gossip overlays (prego). In: 2010 IEEE 10th International Conference on Computer and Information Technology (CIT), pp. 83–90. IEEE (2010)
8. Carlini, E., Ricci, L., Coppola, M.: Reducing server load in mmog via p2p gossip. In: 2012 11th Annual Workshop on Network and Systems Support for Games (NetGames), pp. 1–2. IEEE (2012)
9. Jelasity, M., Babaoglu, O.: T-man: Gossip-based overlay topology management. In: Brueckner, S.A., Di Marzo Serugendo, G., Hales, D., Zambonelli, F. (eds.) ESOA 2005. LNCS (LNAI), vol. 3910, pp. 1–15. Springer, Heidelberg (2006)
10. Baraglia, R., Dazzi, P., Mordacchini, M., Ricci, L., Alessi, L.: Group: A gossip based building community protocol. Smart Spaces and Next Generation Wired/Wireless Networking, 496–507 (2011)
11. Baraglia, R., Dazzi, P., Mordacchini, M., Laura, R.: A peer-to-peer recommender system for self-emerging user communities based on gossip overlays. Journal of Computer and System Sciences (2012)
12. Carlini, E., Coppola, M., Dazzi, P., Laforenza, D., Martinelli, S., Ricci, L.: Service and resource discovery supports over p2p overlays. In: Proc. of ICUMT 2009, Hersonissos, Crete, Greece, pp. 1–8. IEEE Computer Society (October 2009)
13. Marzolla, M., Mordacchini, M., Orlando, S.: Peer-to-peer systems for discovering resources in a dynamic grid. Parallel Computing 33(4), 339–358 (2007)
14. Marzolla, M., Mordacchini, M., Orlando, S.: A p2p resource discovery system based on a forest of trees. In: Proc. DEXA 2006, pp. 261–265. IEEE (2006)
15. Frey, D., Royan, J., Piegay, R., Kermarrec, A., Anceaume, E., Le Fessant, F.: Solipsis: A decentralized architecture for virtual environments. In: Proceeding of 1st International Workshop on Massively Multiuser Virtual Environments (MMVE 2008), pp. 29–33 (2008)
16. Chen, A., Muntz, R.: Peer clustering: a hybrid approach to distributed virtual environments. In: Proceedings of 5th ACM SIGCOMM Workshop on Network and System Support for Games, p. 11. ACM (2006)
17. Bharambe, A., Pang, J., Seshan, S.: Colyseus: A distributed architecture for online multiplayer games. In: NSDI 2006: 3rd Symposium on Networked Systems Design & Implementation, pp. 155–168 (2006)
18. Hu, S., Chang, S., Jiang, J.: Voronoi state management for peer-to-peer massively multiplayer online games. In: 5th IEEE Consumer Communications and Networking Conference, CCNC 2008, pp. 1134–1138. IEEE (2008)
19. Jiang, J., Huang, Y., Hu, S.: Scalable AOI-cast for peer-to-peer networked virtual environments. Journal of Internet Technology 10(2), 119–126 (2009)
20. Ricci, L., Carlini, E., Genovali, L., Coppola, M.: Aoi-cast by compass routing in delaunay based dve overlays. In: 2011 International Conference on High Performance Computing and Simulation (HPCS), pp. 135–142. IEEE (2011)
21. Ramaswamy, L., Gedik, B., Liu, L.: A distributed approach to node clustering in decentralized peer-to-peer networks. IEEE Transactions on Parallel and Distributed Systems 16, 814–829 (2005)
22. Datta, S., Giannella, C., Kargupta, H.: Approximate distributed k-means clustering over a peer-to-peer network. IEEE Transactions on Knowledge and Data Engineering 21(10), 1372–1388 (2009)

23. Jelasity, M., Voulgaris, S., Guerraoui, R., Kermarrec, A.M., Van Steen, M.: Gossip-based peer sampling. ACM Transactions on Computer Systems (TOCS) 25(3), 8 (2007)
24. Voulgaris, S., Gavidia, D., Van Steen, M.: Cyclon: Inexpensive membership management for unstructured p2p overlays. Journal of Network and Systems Management 13(2), 197–217 (2005)
25. Voulgaris, S., van Steen, M.: Epidemic-style management of semantic overlays for content-based searching. In: Cunha, J.C., Medeiros, P.D. (eds.) Euro-Par 2005. LNCS, vol. 3648, pp. 1143–1152. Springer, Heidelberg (2005)
26. Legtchenko, S.: Blue Banana: resilience to avatar mobility in distributed MMOGs. Networks, 171–180 (2010)
27. Dazzi, P., Carlini, E.: Epeerdemics: A peer-to-peer simulator targeting epidemic-based protocols. ERCIM News 2013 (92) (2013)