nSense: Interest Management in Higher Dimensions

Robert Rehner
Databases and Distributed Systems
Technische Universität Darmstadt
Darmstadt, Germany
rehner@dvs.tu-darmstadt.de

Max Lehn
Databases and Distributed Systems
Technische Universität Darmstadt
Darmstadt, Germany
mlehn@dvs.tu-darmstadt.de

Alejandro Buchmann
Databases and Distributed Systems
Technische Universität Darmstadt
Darmstadt, Germany
buchmann@dvs.tu-darmstadt.de

I. Introduction

In Networked Virtual Environments (NVE), *Interest Management* (IM) is a key part of the system that determines which of a participant's actions have to be communicated to which subset of the other participants. Since traditional client/server approaches have several drawbacks, a variety of alternative peer-to-peer (P2P) approaches have been presented by the community [1]. In many of these approaches, interest management is also responsible for maintaining connectivity of the P2P network. Aiming for latency-sensitive applications, systems like VON [2] and pSense [3] use mutual notification mechanisms, building the topology based on the participants' virtual world proximities. A common limitation of these, however, is their fixed dimensionality: most are only designed for two spatial dimensions, a few are capable of handling three.

We argue that adding dimensions to the virtual space in which units are located enables overlay support for specific game features, such as wormholes or portals. Connectivity among peers on both sides of a portal is particularly helpful for enabling fast jumps through the portal. Similarly, unit-type or team visibilities can be modeled as virtual layers on an additional dimension. An overlay that supports additional dimensions can optimize its topology and communication, e.g., reducing neighbor sets to actually visible units. In addition, node properties like link capacity, reliability, reputation, or network coordinates can be modeled as dimensions and used for load balancing [4].

In this work, we discuss options for dealing with an arbitrary number of dimensions. We propose a dynamic localized peer-to-peer IM that supports and exploits *any number of dimensions*. In our IM, peers communicate directly within their vision range. The space outside of the vision range is divided in sectors, each guarded by a sensor node that notifies them of approaching peers. We determine criteria for an efficient sector partitioning, discuss several approaches and present a suitable algorithm.

II. Dealing with More Dimensions

There are different ways to implement IM in a many-dimensional space. One approach is to map the d-dimensional space to $d - x$ dimensions, which can be handled by existing IM solutions. The easiest way is to project the x dimensions onto a $(d - x)$-dimensional hyperplane. A downside of this approach is that it essentially ignores x dimensions. Projected-away dimensions cannot be used for overlay-supported interest filtering, leading to a lower precision.

A similar approach is to reduce d-dimensional space to one dimension using space-filling curves, like the Z-order curve or Hilbert curve [5]. Space filling curves preserve locality to a certain degree. The smaller the range needed on the one-dimensional curve far a given d-dimensional range, the more efficient is the use of the space-filling curve. However, no matter what curve is used, this efficiency inevitably becomes very low in certain situations [5].

Another option is to actually handle all dimensions in the overlay. Almashor and Khalil [4] have proposed to use three-dimensional Voronoi diagrams, employing the third dimension to order the nodes by capacity for load balancing reasons. Generally, Voronoi diagrams can be built for any number of dimensions. The downside of Voronoi approaches for IM, however, is their relative instability in case of high densities and velocities [6].

Our approach is a generalization of the idea of pSense [3]. In pSense, IM is accomplished through a mutual notification system, in which participants communicate directly within their *Area of Interest* (AOI). To get notifications about other participants that enter the AOI, pSense divides the outside of each participant's AOI in eight sectors. In each sector, the node closest to (but outside of) the AOI is chosen as *sensor node*. Therefore, each node maintains up to eight sensor nodes. Sensor nodes are responsible for notifying the participant about new neighbors entering its AOI, as well as about better sensor node candidates.

III. Multi-dimensional Overlay

Generalizing pSense to more dimensions requires answers to mainly two questions: (1) what partitioning scheme can be applied to create sectors around the then d-dimensional sphere, and (2) which number of sectors is necessary for d dimensions. Since a node regularly communicates with each of its sensor nodes to keep track of their positions, it is desirable to keep their number to a minimum. The lower bound is set by the sizes of the resulting sectors. A sector must be small enough that no node can 'slip through' to a node's AOI without being detected by the responsible sensor node. For the selection of the partitioning scheme, we therefore define two basic goals: minimizing (1) the number of sectors and (2) the diameter of the sectors.

The principle of dividing the virtual world in sectors is illustrated in Fig. 1. To determine a sector, we project rays starting from the center of the sphere (where the participant is located) and going through one of the divided areas on the

978-1-4799-0521-8/13/$31.00

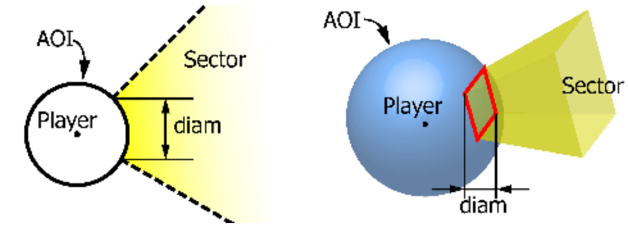

Fig. 1. Dividing the space into AOI and sectors. Left: 2D, right: 3D. $diam$ specifies the diameter of the exemplary sector.

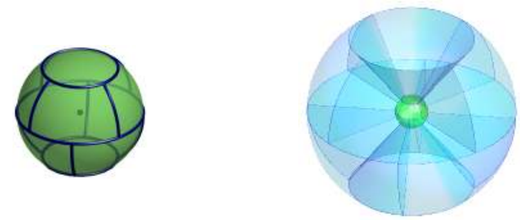

Fig. 2. Left: A three dimensional AOI with the player in the middle divided in 10 regions. Right: A view of the corresponding sectors of this partitioning.

sphere. In 2D we obtain a triangle-shaped sector area; in 3D we obtain a pyramid-shaped sector volume.

The choice of eight sensor nodes for the 2D space in pSense is rather arbitrary: there exists no geometrical proof that eight sensors are necessary or enough for every possible situation, but practical results show that this number seems to work well. Ideally, the sufficiency of a given number of sensor nodes N for a given dimensionality d should be both geometrically proven and practically tested though experiments.

There are several options to partition a sphere in equally sized areas with small diameter [7]. At a first glance most appealing are *convex regular polytopes*; also known as *Platonic solids* in the 3D space. This method, however, is restricted to only a few configurations with fixed numbers of sectors. Especially in dimensions higher than four, only three convex regular polytopes exist.

We decided to use the method by Leopardi [8] because it yields geometrically simple sector shapes (i.e., circles and rectangles). Therefore, determining the sector for a given point can be calculated efficiently. His *recursive zonal equal area partition* algorithm, $\mathrm{EQ}(d, N)$, takes the dimensionality d and the number of desired partitions N as input. The partition is achieved by dividing the unit sphere in two polar caps and several collars with subdivisions. We treat polar caps as collars with only one subdivision. The output consists of the angles corresponding to the caps and collars, the number of collars n and the number of subdivisions for each collar $(m_1, ..., m_n)$, such that $N = 2 + \sum_{i=1}^{n} m_i$. The actual subdivision of collar i is than obtained by applying $\mathrm{EQ}(d-1, m_1)$ recursively until the collar consists of only one partition. Fig. 2 left shows an example of a 3D sphere partitioned in 10 regions.

To determine the sector of a point $\vec{p} = (p_1, \ldots, p_n)$ we need the basis vectors $\vec{e}_1, \ldots \vec{e}_n$ for the space. The algorithm works as follows:

1) determine the angle α between $\vec{p}$ and $\vec{e}_i$ for $i = 1$
2) look-up to which collar α belongs
3) if the collar consists of only one region, we are done
4) for collars with multiple regions, we recursively repeat the algorithm with $i := i + 1$

Pre-computing the sectors with the $\mathrm{EQ}(d, N)$ algorithm is complex, but can be done at build time. In return, determining the sector to a given point at runtime is simple and efficient. The algorithm needs to compute at most $d-1$ angles between p and the base vectors; which can be parallelized. These angles are than compared with the pre-computed sector-angles to determine the sector for the point.

With the partitioning scheme defined, the remaining question is: what is an optimal number of sectors for a given d? We started implementing our approach in the *Planet PI4* testbed [9]. This gaming testbed already contains several other IM algorithms and has the options to run the game in a simulation as well as on the real network. With it, we will investigate the relation between dimensionality (d) and the sector count (N). To lower bandwidth usage, we look for a small N which still produces minimal IM errors. The testbed approach also enables us to compare the performance of using additional dimensions for IM against traditional approaches.

Alongside the empirical evaluation, we plan a theoretical, i.e., geometrical, analysis of the relation between dimension, sector size, and the likelihood of participants entering the AOI without notification by any sensor.

Acknowledgment

This work has been funded by the German Research Foundation (DFG) as part of the CRC 1053 "MAKI" and the GRK 1343 "Topology of Technology".

References

[1] A. Yahyavi and B. Kemme, "Peer-to-Peer Architectures for Massively Multiplayer Online Games: A Survey," *ACM Computing Surveys*, vol. 46, no. 1, 2013.

[2] S.-Y. Hu, J. Chen, and T. Chen, "VON: a scalable peer-to-peer network for virtual environments," *IEEE Network*, vol. 20, no. 4, pp. 22–31, 2006.

[3] A. Schmieg, M. Stieler, S. Jeckel, P. Kabus, B. Kemme, and A. P. Buchmann, "pSense-Maintaining a Dynamic Localized Peer-to-Peer Structure for Position Based Multicast in Games," in *Peer-to-Peer Computing, 2008. P2P'08. Eighth International Conference on*, 2008, pp. 247–256.

[4] M. Almashor and I. Khalil, "Load-Balancing Properties of 3D Voronoi Diagrams in Peer-to-Peer Virtual Environments," *IEEE Parallel and Distributed Systems*, pp. 839–844, Dec. 2010.

[5] P. Ganesan, B. Yang, and H. Garcia-Molina, "One Torus to Rule them All: Multi-dimensional Queries in P2P Systems," in *WebDB '04*. New York, New York, USA: ACM Press, Jun. 2004, pp. 19–24.

[6] C. Groß, M. Lehn, A. P. Buchmann, and R. Steinmetz, "Towards a Comparative Performance Evaluation of Overlays for Networked Virtual Environments," in *Proceedings of the 11th IEEE International Conference on Peer-to-Peer Computing*, IEEE, Ed. IEEE, Sep. 2011, pp. 34–43.

[7] E. Saff and A. Kuijlaars, "Distributing many points on a sphere," *The Mathematical Intelligencer*, vol. 19, no. 1, pp. 5–11, Dec. 1997.

[8] P. Leopardi, "A partition of the unit sphere into regions of equal area and small diameter," *Electronic Transactions on Numerical Analysis*, vol. 25, pp. 309–327, 2006.

[9] M. Lehn, C. Leng, R. Rehner, and A. Buchmann, "An Online Gaming Testbed for Peer-to-Peer Architectures," in *SIGCOMM'11*. ACM, 2011.