On the Use of Structured Peer-to-Peer Systems for Online Gaming

Simon Rieche, Marc Fouquet, Heiko Niedermayer, Klaus Wehrle, Georg Carle
Wilhelm-Schickard-Institute for Computer Science
University of Tübingen, Germany
firstname.lastname@uni-tuebingen.de

***Abstract*— Massively multiplayer games are becoming increasingly popular today. However, even high-budget titles suffer from downtimes because of hard- and software problems. Our approach is to use structured Peer-to-Peer technology for the server infrastructure of massively multiplayer online games, which improves reliability and scalability of these applications.**

I. Introduction

Multiplayer games that can be played over the Internet have become very popular in the last few years. An interesting subcategory are the so-called massively multiplayer role-playing games (MMORPGs) that allow thousands of player characters to share a single game world. Such a world is usually run on a high-performance and high-availability server cluster. However, even with games that have been extensively beta-tested, downtimes of several hours because of hard- or software failures are not uncommon. We propose to use a structured Peer-to-Peer technology for the organization of the infrastructure. Our goal is to make it possible to connect and disconnect infrastructure hosts at runtime. It would also allow to dynamically balance the load on the server infrastructure.

II. Peer-to-Peer based Infrastructure Support

There are a number of technical requirements that our system needs to fulfill. It has to scale and to react to dynamic changes of the load, caused e.g. by temporary hot spots in the game world. Since we want reduce downtimes, it is necessary to support joins and leaves of servers. These requirements may first imply a pure Peer-to-Peer approach. However, since we have to avoid cheating and do authorization and accounting we propose to limit the Peer-to-Peer network to a trusted infrastructure of server peers.

A game world is usually defined as a map. As CAN is a DHT-based on an d-dimensional (e.g. 2 or 3) coordinate space it is best-suited for map-based scenarios. Fig. 1 shows a distribution of servers over the map, which is mapped into a CAN based structured Peer-to-Peer system. Since the zones at the edge of the CAN identifier space are connected with the opposite zones for routing and load balancing, also game worlds simulating a terrestrial globe are possible.

CAN includes a mechanism, that allows neighboring peers to take over the area of a server that has disconnected. However, the state information that was stored about this area would be lost, so this data needs to be replicated somewhere in the network. A peer that owns an area can simply propagate changes to the game state to $2d$ neighbor peers. Parts of this information are also useful for the nodes, as some of players in these areas may be close to the borders and therefore able to see what happens in other areas.

In case of multiple neighbors in one of the $2d$ directions, the information is only replicated to one of these neighbors. The other neighbors still receive information that might be interesting for their players, e.g. information about objects and characters that are within visible range of the zone border. The background zone reassignment algorithm of CAN can be used to reduce fragmentation of the game world.

Load balancing is an important issue, as one can not expect the players to distribute uniformly in the game world. Some places in the game world, like cities for example, can be expected to be more populated on the average. This knowledge could be used to statically configure the servers of a MMORPG in a way that distributes the expected load evenly.

But sooner or later players will do something unexpected. In many games it is common to form large groups for raiding enemy territories for example. In such a case dynamic load balancing is required, e.g. using the Virtual Server approach.

Any Peer-to-Peer approach has to support joins and leaves of nodes. As a consequence, new hosts can be added to the Peer-to-Peer infrastructure when needed. Suitable hosts are not only high-end servers, the game could also run on standard PCs. These PCs could also be located at different geographic locations.

Since the infrastructure peers are controlled by the game company all nodes in the network are trusted. The peers have to be authenticated since an outside peer could try to enter the network and attack it.

III. Conclusions

We have introduced an approach to reduce downtimes and increase scalability in massively multiplayer online-gaming using Peer-to-Peer technology for the game infrastructure.

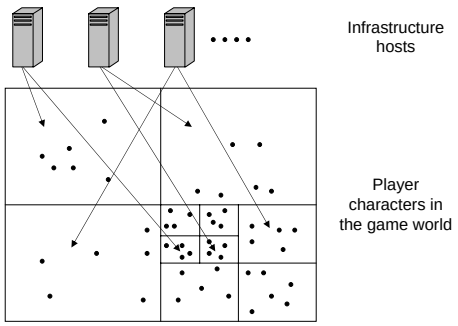

Fig. 1. Architecture of the Peer-to-Peer based Infrastructure