An Online Gaming Testbed for Peer-to-Peer Architectures

Max Lehn Christof Leng Robert Rehner Tonio Triebel Alejandro Buchmann
Databases and Distributed Systems
Technische Universität Darmstadt, Germany
{mlehn,cleng,rehner,buchmann}@dvs.tu-darmstadt.de
Praktische Informatik IV
Universität Mannheim, Germany
triebel@pi4.informatik.uni-mannheim.de

ABSTRACT

In this demo we present a testbed environment for Peer-to-Peer (P2P) game architectures. It is based on Planet PI4, an online multiplayer game whose gameplay provides a standard workload for a set of gaming-specific network interfaces. Its pluggable architecture allows for the evaluation and comparison of existing and new P2P networking approaches. Planet PI4 can run on a real network for prototypical evaluation as well as in a discrete-event simulator providing a reproducible environment.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed Applications; C.4 [**Performance of Systems**]: Design studies

General Terms

Experimentation, Performance, Measurement

1. INTRODUCTION

Massively multiplayer online games (MMOGs) have become very popular in the recent years. Well-known examples are *World of Warcraft* and *Eve Online*, which support up to several ten thousand players simultaneously online in one virtual world. Today, these MMOG use server-based network architectures for the synchronization of the game content among the players. However, the permanent (inter-) activity of the players puts a high demand on the servers' resources, making the operation of an MMOG expensive.

This is why peer-to-peer architectures are an appealing approach to saving server resources. P2P research has already brought up a plethora of P2P systems designed specifically for networked games or networked virtual environments (NVE). But until now, only very few P2P systems have made it to real games. (One example for the latter is Badumna [6].) Many of the academic approaches are evaluated using an oversimplified model of the game and the players' behavior. Round-based simulation and random walk or random waypoint mobility models are simple to use; however, they cover only a small part of the issues of a real game on a real network. Deploying a real game equipped with a new P2P mechanism, on the other hand, is a hard piece of work. Furthermore, measurements may be hardly reproducible in real-world Internet setups.

SIGCOMM'11, August 15–19, 2011, Toronto, Ontario, Canada.
ACM 978-1-4503-0797-0/11/08.

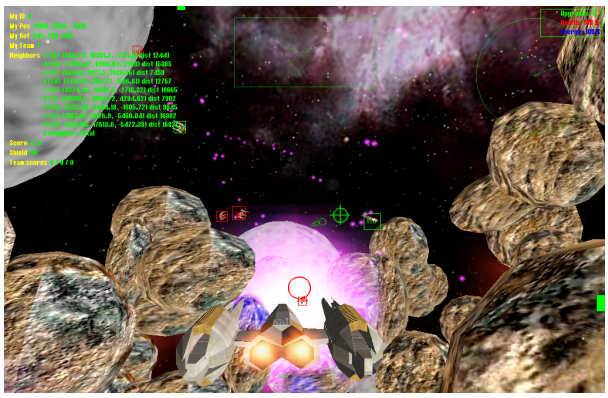

Figure 1: A screenshot of the game Planet PI4

2. GOALS

We are developing the 3D real-time massively multiplayer online game *Planet PI4* [10, 7], which serves as a benchmark for various P2P networking components. For an effective evaluation we have identified the following objectives:

- The gameplay must be complex enough to represent a real game and cover all relevant aspects.
- It should be attractive to real players, which provide the reference player behavior.
- At the same time it should be as simple as possible to allow focusing on the important aspects and to be successfully played by bots.
- It should be able to run in a real network as well as in a deterministic emulated network environment.
- It should be resource-efficient for a good simulation scalability.
- The networking components must have well-defined and flexible interfaces to facilitate the replacement of the network architecture.

We believe that with these features we can provide a testbed for a realistic comparison of P2P network architectures. Supporting execution on a real network ensures that the tested systems do not use any shortcuts available only in a simulation, while network emulation is the only way to achieve a reproducible network environment. A realistic user model for a game can only be obtained by analyzing the behavior of human players, but again, a reproducible workload has to be generated synthetically. Our approach is to collect traces from human players, e.g., by running a client-server game session, from which we derive an abstract user model. This model will be fed in a bot implementation to reproduce human behavior. Using player traces directly as a game workload is infeasible, because the traces are not

parameterizable for scaling the workload. Moreover, traces cannot reproduce the interactivity between players, which is in turn influenced by the network environment.

3. GAMEPLAY

In Planet PI4 each player joins a team to play against other teams. The game set is in an asteroid field, and each player navigates his spaceship and shoots at other ships. Once a ship is destroyed, the player respawns at the team's initial position. The goals are to destroy the other teams' players and to capture bases. Bases are strategic points of interest providing certain improvements (energy, points, etc.) to the possessing team. To capture a base, it is necessary to stay within the range of the base while keeping players of other teams out. Figure 1 shows a screenshot of the game.

The asteroid field defines the three-dimensional gameplay region. Bases particularly attract players, causing hotspots in player density. To evaluate scalability, the game map size and player density must be variable. Therefore, the map is generated algorithmically, based on a common random seed.

4. ARCHITECTURE

The testbed architecture has been consequently designed to fulfill the above defined requirements. Figure 2 shows its high-level components. The interfaces allow for an exchangeability of the components on the different layers. PlanetPI4's game core provides a player control interface which can be used either by the GUI or by bot implementations. The network interfaces connect the exchangeable network parts, and the system interfaces abstract the run-time.

4.1 Network Components

Different networking issues are split into separate interfaces, allowing independent implementations for each.

Spatial Multicast The most important functional requirement for the networking component of a massively multiplayer online game is the dissemination of position updates. In fact, position updates cause most traffic in MMOG [3]. This is also why a large fraction of publications of P2P systems for games focus on this topic, e.g., VON [5], pSense [8], and Donnybrook [1].

Object Management Another important issue is the management of persistent game objects that can be manipulated by players. Exemplary systems in this category have been presented by Hu et al. [4] and Bharambe et al. [2]. We define *active objects* as objects which have an associated 'think function' for object-specific processing.

Channel-based publish/subscribe For team communication as well as object-specific updates we have defined an interface for channel-based publish/subscribe.

Global statistics Player and team scores that are globally available are stored using the global statistics interface.

At the present time we have an implementation of pSense and an approach based on BubbleStorm [9] for position updates. A client-server implementation, which serves as a reference, supports the full set of network interfaces, including object management and publish/subscribe communication.

4.2 System Runtime

To be able run in a simulated network environment, the whole game code uses an event-based design. All components which have to regularly execute a certain task register at the central task scheduler. The task scheduler implementation is exchanged for different runtime environments.

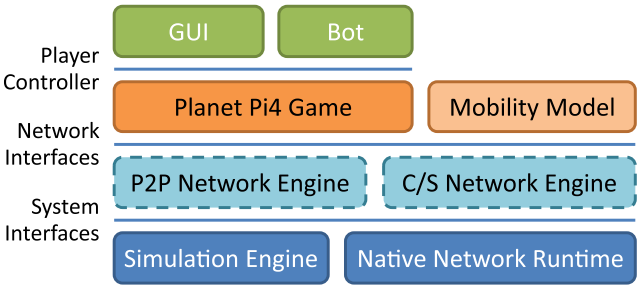

Figure 2: High-level testbed architecture

To run in a discrete-event simulator, the scheduler creates simulator events for each task, while the native-network implementation runs a real-time event loop.

5. CONCLUSION AND OUTLOOK

Our modular testbed provides powerful means for a realistic evaluation and comparison of P2P gaming architectures. Evaluation in both a real environment and an emulated network promotes valuable insights into the properties of the tested systems. Being able to run a real game does not only increase the confidence in the obtained results of a P2P system, but also gives a feeling of its real-world applicability.

As next step we will conduct user studies to obtain player behavior data. From these, we want to derive user models and build bot implementations with a calibrated behavior.

This work was partially funded by the DFG research group FOR 733 and the DFG research training group GRK 1343.

6. REFERENCES

[1] A. Bharambe, J. R. Douceur, J. R. Lorch, T. Moscibroda, J. Pang, S. Seshan, and X. Zhuang. Donnybrook: Enabling Large-Scale, High-Speed, Peer-to-Peer Games. *ACM SIGCOMM Computer Communication Review*, 38(4):389–400, 2008.

[2] A. Bharambe, J. Pang, and S. Seshan. Colyseus: A Distributed Architecture for Online Multiplayer Games. In *NSDI '06*, Berkeley, CA, USA, 2006. USENIX Association.

[3] K. Chen, P. Huang, C. Huang, and C. Lei. Game traffic analysis: an MMORPG perspective. In *NOSSDAV'05*, pages 19–24. ACM, 2005.

[4] S. Hu, S. Chang, and J. Jiang. Voronoi State Management for Peer-to-Peer Massively Multiplayer Online Games. In *CCNC '08*, pages 1134–1138, 2008.

[5] S.-Y. Hu and G.-M. Liao. Scalable Peer-to-Peer Networked Virtual Environment. In *NetGames '04*, pages 129–133, 2004.

[6] S. Kulkarni, S. Douglas, and D. Churchill. Badumna: A decentralised network engine for virtual environments. *Computer Networks*, 54(12):1953–1967, 2010.

[7] M. Lehn, T. Triebel, C. Leng, A. Buchmann, and W. Effelsberg. Performance Evaluation of Peer-to-Peer Gaming Overlays. In *IEEE P2P '10*, 2010.

[8] A. Schmieg, M. Stieler, S. Jeckel, P. Kabus, B. Kemme, and A. Buchmann. pSense - Maintaining a Dynamic Localized Peer-to-Peer Structure for Position Based Multicast in Games. In *IEEE P2P '08*, pages 247–256, 2008.

[9] W. Terpstra, J. Kangasharju, C. Leng, and A. Buchmann. BubbleStorm: Resilient, Probabilistic, and Exhaustive Peer-to-Peer Search. In *ACM SIGCOMM '07*, pages 49–60, 2007.

[10] T. Triebel, B. Guthier, R. Süselbeck, G. Schiele, and W. Effelsberg. Peer-to-Peer Infrastructures for Games. In *NOSSDAV '08*, pages 123–124, 2008.