Exploiting P2P in the Creation of Game Worlds

Daniel Hughes
Lancaster University
Lancaster
Lancashire, UK
+44 (0)1524 510351
hughesdr@comp.lancs.ac.uk

Kiel Gilleade
Lancaster University
Lancaster
Lancashire, UK
+44 (0)1524 510339
gilleade@comp.lancs.ac.uk

James Walkerdine
Lancaster University
Lancaster
Lancashire, UK
+44 (0)1524 510352
walkerdi@comp.lancs.ac.uk

ABSTRACT

Peer-to-peer networks are a promising platform for supporting entirely decentralized, distributed multi-user gaming; however, multi-player games typically require highly predictable performance from the underlying network. This is at odds with the inherently unreliable nature of peer-to-peer environments. Existing approaches to providing peer-to-peer support for multi-player gaming focus on compensating for the unpredictability of the underlying network. We propose that rather than trying to compensate for these factors, they can be exploited together with information about the peer-to-peer network in order to address the problem of maintaining a novel gaming experience in the absence of a central authority. In order to explore our proposition, we model the measurable properties of P2P networks within a distributed multi-player game – *NetWorld*. We do this in such a way that the heterogeneous and unpredictable nature of the peer-to-peer environment becomes a positive part of the player's experience.

Categories and Subject Descriptors
J.M [Computer Applications]: Miscellaneous.

General Terms
Measurement, Design, Human Factors.

Keywords
Virtual Reality, Peer-to-Peer, Games, Presence

1. INTRODUCTION

Peer-to-peer (P2P) networks are a logical choice for supporting distributed multi-user virtual worlds as expensive servers are not required. P2P networks also provide good support for the spontaneous formation of ad-hoc groups. Existing approaches to supporting virtual worlds on P2P networks tend to focus upon compensating for the unpredictability of the underlying network [1]. We suggest that, just as the properties of mobile devices have been harnessed to provide novel game-play [2], the properties of P2P networks, specifically their heterogeneity and unpredictability can also be exploited to provide a novel user experience, thus turning P2P networks' perceived weakness for supporting distributed gaming into a strength.

We first looked at the common properties of P2P systems and for ways that these properties might be used to inform the design of a game-world, such that the design of the game-world will reflect the state of the nodes and network links which compose the peer-to-peer network. Based upon these observations, we have developed a proof-of-concept implementation. '*NetWorld*' is a three-dimensional first-person exploration game wherein the virtual game-world is spontaneously generated from the measurable properties of the underlying P2P network.

In order to expedite the development of *NetWorld*, it was developed as a plug-in for Lancaster's P2P Application Framework [3]. The P2P Application Framework provides an abstract layer specifically geared to rapid P2P application development, reducing the burden upon the developer to understand the underlying P2P technologies, thus allowing developers to focus on application development.

The remainder of this document is structured as follows: Section 2 gives background information on the subject area, section 3 describes the properties of P2P systems, section 4 describes *NetWorld*, section 5 describes the P2P Application Framework and section 6 discusses future work.

2. BACKGROUND

Relevant work exists in the fields of distributed virtual worlds and distributed multiplayer gaming. This is discussed in sections 2.1 and 2.2 respectively.

2.1 Distributed Virtual Worlds

A number of projects have looked into using P2P networks to support distributed virtual worlds, however, most of these systems conform to a semi-centralized [4] model of interaction, which relies upon the use of central server(s) to maintain consistency [5]. This is undesirable as it requires the provision of expensive server equipment by a central authority and thus eliminates the possibility of gaming between ad-hoc groups, effectively eliminating the primary benefits offered by using P2P technology to support distributed gaming. Recently, work has been conducted into implementing distributed virtual worlds on fully decentralized P2P networks [6], wherein the virtual world is distributed between peers. In this implementation, users are required to generate zones of the virtual world based upon a set of common rules. While this approach is highly scalable, research has shown that environments generated in this manner typically lack heterogeneity, quickly becoming predictable and thus ceasing to entertain users [7]. Unpredictability and heterogeneity are two factors present in abundance in P2P environments. Indeed, they are typically considered to be serious weaknesses of P2P technologies!

NetWorld extends a fully distributed virtual world implementation by using the unpredictable and heterogeneous properties of the underlying P2P network, together with information about that network to inform the design and evolution of the virtual world with the aim of improving the users' experience as described in section 2.3.

2.2 P2P Game Support

The majority of approaches to supporting distributed, multiplayer gaming on P2P networks, such as [1] and [9] focus upon providing support for real-time communication and consistency management by compensating for the heterogeneous network technologies and local platforms used in the underlying P2P network.

Real Tournament [1] aims to support real-time gaming through the implementation of content filtering schemes on the underlying P2P network, similar to those implemented in the CAN overlay network [8]. In this model, filters of interest ensure that communication of game-state between nodes is bandwidth efficient and furthermore, that game-space is intelligently partitioned between nodes based upon their position in the game world.

Solipsis [9] uses a P2P network to maintain a virtual world that is capable of handling very large numbers of players and game-world objects. Solipsis does this using a P2P network whose architecture corresponds to a Gnutella-like network structure [10]. *Solipsis* optimizes this network by collocating peers which are close in the game-world on the P2P network, thus reducing the distance that messages must travel on the underlying P2P network.

Unlike the above approaches, which attempt to compensate for the heterogeneous network technologies and local platforms used in the underlying P2P network, *NetWorld* seeks to create a novel gaming experience from exactly these properties by making them an integral part of the gaming experience.

2.3 Our Approach

In the *NetWorld* model, the game-world reflects the state of the underlying P2P network. This is the opposite of the approach taken by *Solipsis,* which modifies the structure of the underlying P2P network to reflect the state of the game and distinctly different to the approach used in *Real Tournament,* which hides the heterogeneity and dynamism of the underlying network from players. The way in which *NetWorld's* game-world is generated is somewhat similar to generative games, wherein the game world is created and modified based upon a set of hard-coded rules, however, *NetWorld* adds presence by modeling the real-world network, peers and users. In this way, *NetWorld* seeks to maintain the appeal of the game in a way that generative games cannot. A detailed description of *NetWorld*'s game-play is provided in section 4.

3. PROPERTIES OF P2P SYSTEMS

P2P systems are typically highly heterogeneous both in terms of the local platforms which compose them and in terms of the network links which connect peers. Network links between nodes may range from low-bandwidth, unreliable, mobile links, such as GPRS and GSM, to high-speed, reliable, connections such as wired Ethernet. Furthermore, the local resources of the nodes which compose these networks may range from mobile devices such as smart-phones, with limited memory and computational resources, to modern high-specification workstations. Despite this high level of heterogeneity, all peers and network links share a number of common and quantifiable properties, which are used by *NetWorld* in the creation and evolution of the game world.

Common quantifiable properties of the *nodes* which *compose P2P networks* include:

- *CPU performance*
- *Storage Media Size*
- *Storage performance*
- *Available memory*

Common quantifiable properties of *application level network links* include:

- *Throughput*
- *Loss*
- *Delay*
- *Hop-count on the underlying network.*

The state of these properties varies dynamically, for example the throughput of an application level network link will vary based upon the overall load on the underlying network connection. Similarly CPU performance will be dependent upon the number and nature of competing processes that may be running on a peer. We extended the P2P Application Framework [3] to monitor each of the eight properties discussed above, allowing them to be modeled in virtual game-world's. While it may be possible to utilize other properties, we have aimed for a universal set that is common to all nodes, potentially including devices with very restricted local resources such as smart-phones. Monitoring of node and link resources is performed by the P2P Application Framework software [3], an instance of which runs on each peer participating in the P2P network. Local peer and link resources cannot be measured directly in Java as processes that run within the Java Virtual Machine (JVM) are separated from the underlying system by a layer of abstraction. Therefore, where possible (CPU performance, storage performance, throughput, loss and delay), these properties are measured indirectly using custom Java benchmarking tools. Where this was not possible (available memory and hop-count on the underlying network), these properties are measured directly by using JNI to interface with the underlying platform (e.g. Windows or Linux). All monitoring is performed locally at each peer on a periodic basis and reported via the P2P application framework to the network's distributed indexing servers (see Section 5). This allows the current state of each peer on the network to be discovered by any other peer, as described in [3]. Along with the state of each node, each indexing server also maintain a representation of the current network graph, showing the active connections between peers. These two data sets are used together to create the game-world map and room properties as described in section 4.

4. NETWORLD

NetWorld is a three-dimensional first-person exploration game composed of an extensible number of rooms. The game-world closely correlates with the underlying peer-to-peer architecture in order to foster identification by the player between the game-world and the real-world which it relates to. Each room in *NetWorld* corresponds to a peer on the underlying P2P network and each connecting corridor represents an active network link. Each room is hosted by a different peer on the underlying network as illustrated in Figure 1. Players may only access rooms hosted by the neighbors of the peer whose zone they are currently traversing and access is only possible when the corresponding P2P network link is active. In this way, the game-world directly reflects the underlying P2P network. For example, the network visualization shown in Figure 1 illustrates that Peer 'Cloud' only has active network links with 'Danny' and 'Kev'. Therefore a player traversing the room hosted by 'Cloud' will only have access to these two zones. Each player in *NetWorld* maintains a customized puzzle based upon the 'lights out' [12] game, which can be played by any peer that accesses this peer's room. As peers explore the game world, they have the opportunity to collect points by solving these puzzles. Player high scores are held by the P2P Application Framework on distributed indexing server(s), located throughout the network [3]. These scores are made visible to all players during the game. There is no end-goal to *NetWorld*. Players are expected to continually explore the changing environment and compete against a changing user community to amass the most points.

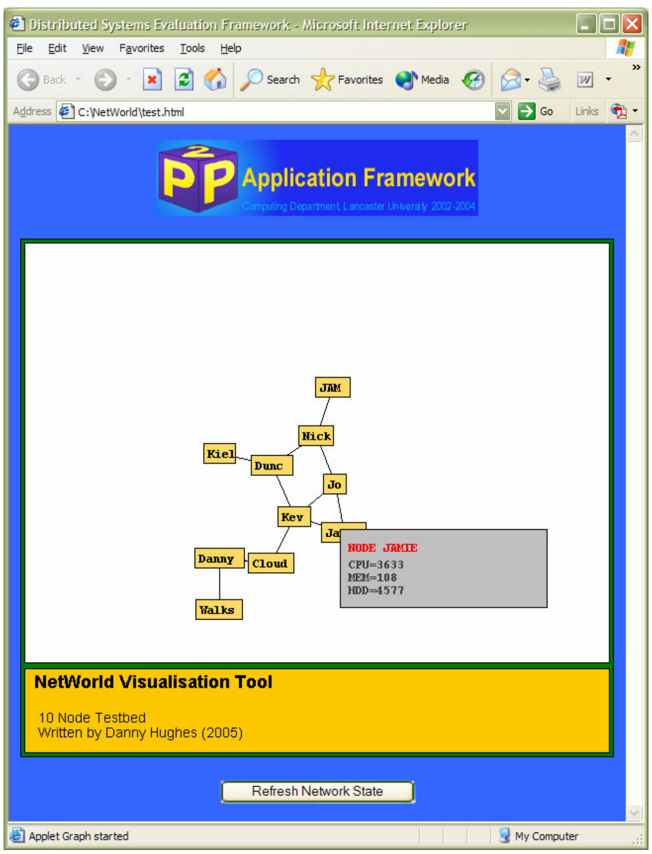
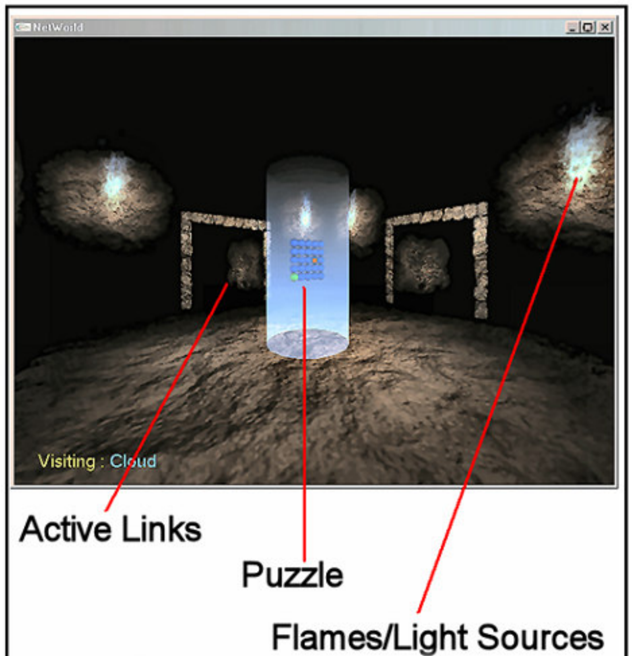


Figure 1: NetWorld Visualization Tool

The current P2P network structure may also be visualized by means of a Java Applet hosted by each of the indexing servers. This is shown in Figure 1. Each box represents a peer (room) and each arc represents a network link (corridor). The current state of each peer along with their relative position can be viewed via the applet, providing a navigation tool for players and a more formal visualization of the current network state. Within the game, we used the properties of CPU power, available memory, storage performance and hop-count to influence the design of entities within the game-world. Figure 2 shows a screenshot of *NetWorld* and highlights how the properties of the underlying network were to influence the appearance of the game world.



P2P Property	Game-World Property
CPU Performance	Flame Ferocity / Puzzle Time Limit
Available Memory	Flame Colors / Puzzle re-try attempts
Storage Performance	Flame Density / Score Multiplier

Figure 2: Modeling P2P Properties

As can be seen in Figure 2, each 'lights out' game is customized according to the peer's system information, such that CPU speed is used to control puzzle time-limit, available memory determines how many times the puzzle may be replayed and storage media performance determines the number of points that will be awarded for successful completion of the puzzle. This information is also manifested in environmental effects within the room, which are used to indicate both the state of the peer hosting the room and also the type of game that a player may expect if they choose to try and solve the puzzle. Players may also 'flip' the environmental effects at any time, to instead represent information about each neighbor peer, thus aiding players in navigation. The rendering of the game-world occurs entirely on the local peer, based upon state information received from the P2P framework. Information exchange occurs at the point when a player enters a new room. Our current implementation is static, such that the properties reported upon entering a room persist as long as a user remains in that room. This means that while the geography of the game-world may change the experience within each room remains consistent.

NetWorld benefits from peers choosing to remain online while they are not exploring the game-world. When a user leaves the system, the room they are hosting, the contained puzzle and associated real-world presence will disappear from *NetWorld*. This is detrimental to users who are currently playing the game and for this reason *NetWorld* supports game-play in both a *passive* or *active* mode. These two ways of playing *NetWorld* are compared below:

- **Active Game Play**
  As peers navigate the game world, they attempt to collect points by solving puzzles. As these puzzles are set by real-world individuals, this encourages competition between players. However, as the network graph (and hence the geography of the game-world) is constantly changing, finding the room owned by a peer whose puzzle a user wishes to solve is challenging. Reaching a room may necessitate waiting for appropriate links between peers to be established followed by rapid navigation of the game-world.

- **Passive Game Play**
  As *active* players explore *NetWorld* attempting to solve puzzles, those peers that are online, but not actively using the game participate *passively*. Every time a peer fails to solve one of the customized puzzles, the player who set that puzzle gains points. The idea being that players will take care when constructing puzzles in order to outsmart their peers. The possibility to earn points by remaining online is also an incentive for users to remain part of the community.

In the case of either *active* or *passive* game-play, presence enhances the player's experience. Examples of this include the identification of personalized puzzles with users, the manner in which the geography of the game-world reflects real world entities and the way in which the condition of rooms and corridors reflect the state of those entities.

5. THE P2P APPLICATION FRAMEWORK

NetWorld is underpinned by Lancaster's P2P Application Framework [3], which is designed to facilitate the rapid development of P2P applications. It achieves this by providing a simple application-centric API which abstracts over the underlying P2P technologies. The services provided by the framework are derived from common functionality identified within existing P2P applications. This includes point-to-point message passing, search and resource transfer [3].

All applications are developed as Java 'plug-ins' for this framework based upon a specified programming interface. As such, they are independent of the underlying P2P substrate. This flexibility is particularly appealing in the context of *NetWorld*, as the game could be deployed across a variety of P2P substrates, which, as they are reflected in the resulting game-world, will result in distinctly different gaming experiences. The architecture of the P2P Application Framework is illustrated in Figure 3.

The *NetWorld* plug-in registers two resource types; peer status information and a user-customized lights-out puzzle. This allows other plug-ins to identify and make use of these resources. For example peer status information may also be used by a visualization plug-in, or the puzzles provided by users could be accessed directly, rather than as part of the *NetWorld* game.

The P2P Application Framework is currently implemented in Java, running over the JXTA [11] P2P substrate, however, Java is not well suited to supporting 3D first-person games such as *NetWorld*; therefore, the graphical parts of *NetWorld* were implemented using C and OpenGL. This interfaces with the P2P application framework through a Java Native Interface (JNI) back-end, which allows it to interface with the P2P application framework.

As in all P2P systems, nodes running the P2P Application Framework may fail without warning. In the P2P Application Framework model, dealing with peer failure is the responsibility of the plug-in applications. In the case of *NetWorld*, interaction between peers only occurs during the transition between rooms; therefore dealing with peer failure is trivial: If a peer fails at any point during room transition, the next closest peer on the network is used instead.

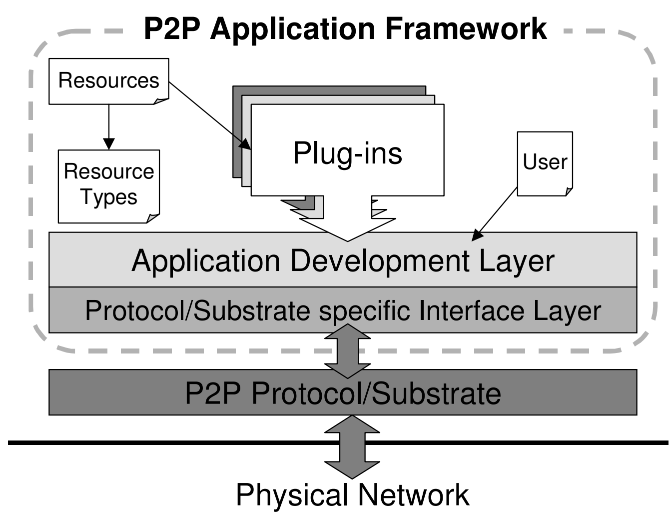


Figure 3: The P2P Application Framework

6. FUTURE WORK

Significant extensions and revisions are planned for *NetWorld*. Our immediate work will focus upon: Addressing the lack of support for multiple players within a single, consistent game-world, refining the mapping of P2P properties onto elements of the virtual world and evaluating the impact of *NetWorld* on the underlying network technologies. These issues are discussed in detail below.

6.1 Multi-Player Support

In its current form, *NetWorld* is not *truly* multi-player. While different players may coexist within the same game-world, these players do not interact; they are effectively transparent to each other. We are currently exploring the possibility of extending *NetWorld* to support multiple interacting players within a consistent game-world; however, this raises several non-trivial issues arising from the difficulty of maintaining consistency in a game-world that is handled by multiple distributed entities. This is particularly difficult in typical P2P environments, wherein peers may join, leave or fail in an unpredictable manner. Furthermore, representing player movement requires a level of real-time communication which is difficult to ensure over a heterogeneous P2P network as peers may have vastly differing local resources and network connections. These issues have been tackled in depth by other work, for example [1] and [9] and we are currently investigating applying such approaches to the *NetWorld*.

6.2 Modeling P2P Properties

The mapping of P2P properties onto properties in the game world is currently rather arbitrary. The current mappings were developed simply to demonstrate that it is possible to meaningfully influence the design of the game-world based upon the properties of the underlying P2P system. However, all properties of the underlying P2P network are not equally suitable for controlling elements in the game world. These properties vary in terms of their range (for example the range of available CPU power is much greater than the range of hard drive performance) and properties also vary in terms of their dynamism (for example available CPU power varies more dynamically than hard drive performance). Thus properties which are highly-dynamic may not be suitable for influencing some features of the game world and similarly, relatively static properties may be unsuitable for influencing others. The visualization chosen for *NetWorld* is also arbitrary. While a maze-like environment wherein peers represent rooms and network links represent corridors intuitively matches a peer-to-peer network, peers could instead be responsible for a section of any kind of game-world.

6.3 Impact on the P2P Network

P2P technologies are particularly useful in mobile gaming environments due to P2P technologies inherent support for the formation of ad-hoc groups, such as discussed in [1] and [2]. Unfortunately, such environments typically offer only low-bandwidth, connections. Therefore the impact of *NetWorld* upon the underlying network infrastructure must be carefully considered. We are currently in the process of testing *NetWorld* in such restricted mobile environments and evaluating the impact of our approach upon the underlying network technologies.

7. ADDITIONAL AUTHORS

Additional authors: John A. Mariani (Lancaster University), email: jam@comp.lancs.ac.uk

8. REFERENCES

[1] McCaffrey D., Finney J., The need for real-time consistency management in P2P mobile gaming environments, Proceedings of the First ACM International Conference on Advances in Computer Entertainment Technology (ACE 2004) in co-operation with SIGCHI, 3-5 June 2004, Singapore.

[2] Cheok A., Fong S., Goh K., Yang X., Liu W., Farbiz F., Human Pacman: A Mobile Entertainment System with Ubiquitous Computing and Tangible Interaction, Proceedings of the Fifth International Symposium on Human Computer Interaction with Mobile Devices and Services

[3] Walkerdine, J., Melville, L., Sommerville, I., A Framework for P2P Application Development, Technical Report COMP-004-2004, Computing Department, Lancaster University, 2004.

[4] Walkerdine J., Melville L., Sommerville I., Dependability Properties of P2P Architectures published in the proceedings of the 2nd IEEE International Conference on Peer-to-Peer computing (P2P'02). Linköping, Sweden, September, 2002.

[5] Gehermann H., Christensen N., TerraPeer – a DVE Architecture and Implementation, Academic Thesis, Informatics and Mathematical Modelling, Technical University of Denmark, 2004.

[6] Gupta S., Kaiser G.., P2P Video Synchronization in a Collaborative Virtual Environment, Technical Report CUCS-008-05, Computing Department, University of Columbia, 2005.

[7] Gilleade K., Dix A., Using Frustration in the Design of Adaptive Videogames. Proceedings of ACE 2004, Advances in Computer Entertainment Technology, ACM Press.

[8] Ratnasamy S., Francis P., Handley M., Karp R., and Shenker S.. A Scalable Content Addressable Network. In Proceedings of ACM SIGCOMM, 2001.

[9] Keller J. and Simon G., Toward a Peer-to-peer Shared Virtual Reality. In 22nd International Conference on Distributed Computing Systems (ICDCS02), pages 595 -601. IEEE Computer Society, Jul 2002.

[10] The Gnutella Protocol Specification, http://dss.clip2.com/GnutellaProtocol04.pdf

[11] Project JXTA, P2P API, Sun Microsystems Inc. More information can be found at http://www.jxta.org/

[12] The 'Lights Out' Game, http://www.haar.clara.co.uk/Lights/