Research Project Proposal

ASCEND

(Adaptive Scalable Cooperative Environment for NVE Developments)

The Design and Implementation of
P2P-based 3D Networked Virtual Environments

(December 2005)

Abstract

ASCEND is a collective research efforts to build *scalable, affordable, and accessible peer-to-peer (P2P)-based 3D networked virtual environments* (NVEs) where participants may share data (e.g. files, photos, blogs) and interact with one another in real-time. ASCEND is based on the highly scalable and topology-consistent *Voronoi-based overlay network (VON)* and we will investigate a number of important P2P-based NVE (P2P-NVE) research issues, including: 3D streaming for P2P-NVEs, state consistency in P2P-NVEs, persistent storage designed for P2P-NVEs, and suitable P2P overlays specifically for NVE applications. ASCEND's motivation is based on three important Internet trends: the rising popularity of *Massively Multiplayer Online Games* (MMOGs), the scalability potential of P2P networks, and the rapid growth of user-generated contents such as blog, file and photo sharing. We hope that ASCEND may bridge the transitions toward the next level of Internet communication that is 3D-based, highly interactive, and user-empowering. ASCEND covers three major research topics in the duration of three years:

1) **3D Streaming for P2P-NVE**: We will investigate how real-time streaming delivery of 3D contents (geometric meshes and textures) may be done in a scalable manner by using P2P networks. We intend to devise distributed visibility determination and parallel streaming methods in order to relieve server load and to shorten client-side download time.

2) **State Consistency Control in P2P-NVE**: We will investigate how the synchronization of object states and event ordering may be supported in a distributed P2P environment by assigning ownerships for each object within a NVE. We intend to adopt a fully-distributed consensus-based voting mechanism for the efficient and correct determination of object owners. An update propagation mechanism that provides high availability of object states in the face of a dynamically changing environment will also be investigated.

3) **Persistent Storage Design for P2P-NVE**: We will investigate how persistent data may be supported in a P2P environment by using only a light-weight server for the eventual data backup. We intend to devise leader node election protocols and data aggregation techniques to support scheduled backups of object states without causing server overload.

Implications of the ASCEND project include an improved understanding of how to construct NVE systems using the P2P paradigm, the creation of a set of open source programming libraries to support NVE developments, and the training of researchers and developers proficient in the new research field of P2P-based NVE systems.

Keywords: peer-to-peer (P2P), networked virtual environment (NVE), scalability, Voronoi diagram, overlay network, 3D streaming, consistency, distributed storage, neighbor discovery, online games.

Introduction

We propose the research project *ASCEND (Adaptive Scalable Cooperative Environment for NVE Developments)* that aims to build *scalable, affordable, and accessible peer-to-peer (P2P)-based 3D networked virtual environments* (NVEs) where participants may share data (e.g. files, photos, blogs) and interact with one another in real-time. We will study a number of important P2P-based NVE (P2P-NVE) research issues which include: 3D streaming for P2P-NVEs, state consistency control in P2P-NVEs, persistent storage design for P2P-NVEs, and suitable P2P overlays specifically for NVE applications. ASCEND's motivation is based on three important Internet trends: the rising popularity of Massively Multiplayer Online Games (MMOGs), the scalability potential of P2P networks, and the rapid growth of user-generated contents such as blog, file and photo sharing. We hope that ASCEND may bridge the transitions toward the next level of Internet communication that is 3D-based, highly interactive, and user-empowering.

Networked virtual environment (NVE) [Singhal and Zyda 1999] is a virtual environment in which multiple users interact with each other in real-time via network connections, even though they may be physically located around the globe. It can be regarded as a distributed system providing immersive multiuser experience in a shared syntactic world. Users situated at different physical locations may meet and interact through 3D representations of themselves (called *avatars*). The study of NVE began with U.S Department of Defense's *Simulator Networking (SIMNET)* Project [Miller and Thorpe 1995] in the 1980s, and was researched and developed by DARPA, the same agency that created the Internet. NVE has since been growing steadily into diverse applications that include military training, scientific visualization, education and entertainment. Arguably the most successful subset of NVE today is what is known as *Massively Multiplayer Online Games* (MMOGs), where thousands to *hundreds of thousands* of players interact with each other simultaneously in a virtual universe. MMOG has become one of the fastest growing genres of computer games since its inception in the mid 1990's, and is now a popular form of entertainment in North America and Asia, especially among the youth. As of 2005, 4 millions paid subscribers are reported for the MMOG *Lineage* [Lineage 2005], made by Korean game company NCSoft, while *World of Warcraft* [WoW 2005], a game by U.S.-based Blizzard Entertainment, has accumulated over 2 millions subscribers within a year since its release in 2004. In some way, MMOG has become not just a multi-billion industry [IGDA 2004], but a cultural phenomenon as well, and its main attraction has gone beyond gaming to the support of social interactions. With continuing advances in CPU, broadband network, and consumer-grade 3D accelerators, it is not unreasonable to expect MMOG to become an immersive interaction *medium* that may develop beyond entertainment into areas such as education, training, e-commerce, and communication. We may thus be expecting the emergence and widespread use of NVE systems on the Internet in the near future.

However, for NVEs to be truly widespread, they will need to become more scalable, affordable and accessible. Good *scalability* allows a single system to accommodate dynamic user size without service interruption, increasing the service-quality and immersiveness of the user experience. *Affordability* lowers the entry barrier in creating and hosting such systems, while *accessibility* indicates the ease of access of such systems to the ordinary computer user. The ideal scenario is that NVEs may *become scalable to millions of concurrent users and accessible to the general public, yet affordable enough for individuals to create and host*. Indeed, a great example is the development of WWW, whose rapid rise of popularity can be traced to its scalability, affordability, and accessibility.

We can only imagine the impact when NVEs become as popular and widespread as today's WWW. Yet various indicators and hardware trends show that a 3D-based, interactive Internet experience is realizable in the foreseeable future. For example, Microsoft's next OS *Windows Vista*

[MS 2005] contains a 3D interface, 2004 Turing Award winner and Xerox PARC founder Alan Kay has also been leading the research of a distributed 3D Operating System called *Croquet* since 2000 [Croquet 2005].

Existing server-based NVE architectures have inherent scalability limits and are expensive to deploy or maintain [Hu et al. 2005]. While over 150,000 concurrent users have been reported for a single MMOG, the infrastructure cost of scaling user size into the range of millions might become unbearably expensive, without even considering the design and maintenance difficulties. A promising alternative is the *peer-to-peer* (P2P) architecture that has become well-publicized with file-sharing applications such as Napster, Gnutella, Kazza, eDonkey, and VoIP application such as Skype. P2P networks promise high scalability and affordability by self-organizing individual computer nodes into a collective platform where resources (i.e. processing power and bandwidth) are shared. P2P-based NVE systems may also be seen as a distributed systems with self-stabilizing properties, which makes them an interesting topic to study. However, issues such as consistency, persistency, and security will need to be reconsidered when NVEs are adopted to P2P architectures.

The current model of NVE content distribution, which requires users to pre-install proprietary software clients and large amount (hundreds to thousands of megabytes) of content data, also makes the use of NVE inconvenient and inaccessible to the general public. From the experience of WWW, we may infer that for NVE systems to be universally adopted, their contents (e.g. static data such as *3D meshes* and *textures*, or transient data such as user commands or voice chats) will need to be delivered *in real-time* to users quickly and efficiently using just a single client software with standardized data formats and transmission protocols.

A number of P2P-based NVE research have been proposed in the last three years, mainly addressing the basic question of constructing a P2P overlay. Among existing approaches [Keller and Simon 2003; Knutsson et al. 2004; Kawahara et al. 2004; Hu and Liao 2004], *Voronoi-based Overlay Network* (VON) [Hu et al. 2005] appears to be very promising, with demonstrated scalability, topology consistency, and simplicity. ASCEND aims for a comprehensive study on various issues related to viable NVEs based on VON. General NVE design criteria such as *consistency, responsiveness, scalability, reliability, and persistency* will be thoroughly studied via specific research topics described in the next section. *The main goal of ASCEND is to derive better understandings on how to design NVE systems using the P2P paradigm*. Specifically, we seek to study how 3D contents may be delivered effectively on P2P overlays (i.e. *3D streaming* on P2P networks), and how persistent data and consistency control may be maintained. Secondary goals include the implementation of a prototype P2P-based NVE system for educational use and a related set of open source programming libraries (the *ASCEND library*). We note that *security* is also a very important and relevant issue to many NVE systems, however, as it can be highly application-specific (i.e. solutions likely will not be general), yet at the same time, NVE systems without strict security functionalities may still gain wide appeal (e.g. for distance-learning, simulation, or promotion purposes), we will leave security issues as future research topics.

In some sense, the main relevance of ASCEND is the understanding and development of the next level of Internet communication beyond World-wide Web. While WWW has been an amazing medium for the capture, sharing, and access of *information*, it is a flat and asynchronous medium that does not effectively enable, capture, or deliver the *real-time interactions* among people. Yet real-time interactions of various forms are at the heart of human societies and activities. Evidences are abundant in the popularity and growth of today's mobile phones, instant messaging, and MMOG applications. With continuing hardware trends in CPU, broadband, and 3D acceleration, as well as the growing amount of interests, tools, and industries related to both the creation and consumption of 3D contents (e.g. Computer-aided designs (CAD), TV commercials, computer-synthesized movies, and MMOG), the time is approaching for various *3D-based real-time interactive* *environments* to emerge on the Internet. We believe that the emergence of potentially *millions* of interactive virtual worlds is a matter of time, not viability. These virtual worlds will likely create dramatic impacts on the ways we learn, work, and socialize. Yet the widespread and adoption of virtual environments will require viable architectures that are both scalable and affordable, hence the goal of ASCEND: *to study the main architectural issues involved in constructing P2P-based NVE systems that are highly scalable yet affordable and accessible.*

From a more practical perspective, fundamental understandings in constructing viable P2P-based NVE systems will allow us to speed up the developments and deployments of such systems, creating opportunities for NVE to be applied in promising new areas such as education, training, and social interactions -- fields which so far cannot afford the large budgets to develop and deploy NVE-based solutions as well as the game industry.

Related Work in P2P-based NVE

The main issue in P2P-based NVE is how the entire virtual world may be maintained in a distributed fashion by various participating nodes through connections with only a limited set of neighbors (i.e. how a P2P topology can be maintained through certain node connection patterns). In contrast to file-sharing P2P, where *content search* is the main problem and can be difficult (as a desired content may be located *anywhere* on the network), the desired content in P2P-based NVE is simply the event messages generated by each node's nearby neighbor within its visibility (i.e. an *area of interest*, or AOI, see Figure 1). The general content discovery problem in P2P thus transforms to a *neighbor discovery problem* in the context of P2P-based NVE. Three main approaches have been proposed to deal with the neighbor discovery problem: Distributed Hash Table (DHT)-based, neighbor-list exchange, and mutual notification.

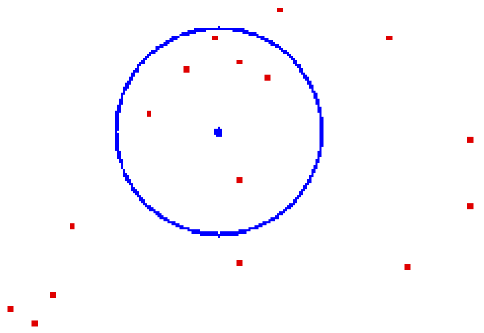

Figure 1. Abstraction of a NVE. Participating users may be represented as *nodes* (the dots). The circle is the *area of interest* (AOI) of the center node.

SimMud [Knutsson 2004] is representative of the DHT-based approach, where the entire virtual world is sub-divided into fixed-size *regions*, each managed by a super-node. Super-nodes maintain contacts with one another through the use of DHT. DHT-based approaches, however, face the problem of high latency due to message relay, thus may not fulfill the real-time requirement of NVE. Fixed-size regions also cannot handle situations when many users have *crowded* into the same region, potentially slowing down or overloading the responsible super-nodes.

The second approach (i.e. neighbor-list exchange) requires each node in the P2P network to maintain a fixed number of nearby neighbors (in terms of distance within the virtual environment), and exchange the “neighbor-lists” regularly so that new neighbors can be learned. However, much of the exchange contains redundant information, as neighbor-relationships do not change rapidly. A more serious issue is that simulations have shown that the P2P overlay may *partition* into mutually unaware parts after some time, as the clustering of nodes may make connecting only to nearest neighbors inadequate in keeping the topology globally connected.

The third approach (i.e. mutual notification) also requires each node to maintain a certain number of neighbors, but neighbor discovery is done in a more message-efficient manner through collaborative notifications between connected nodes. New neighbors are learned from notification messages only when the appropriate situation occurs. As no fixed region is defined and no super-nodes are used, proposals in this category may handle user crowding better than DHT-based approaches. Solipsis [Keller and Simon 2003] and VON [Hu et al. 2005] are the two main proposals, and differ mainly in the types of neighboring nodes to maintain. Solipsis requires each node to be within the outermost boundary of connected neighbors (i.e. within a *convex-hull* formed by the neighbors, see Figure 2), while VON requires all nodes to maintain at least the *enclosing neighbors* as defined by a Voronoi diagram of connected neighbors (Figure 3). Under certain circumstances, Solipsis may not discover neighbors properly and the topology will need to be fixed by a time-consuming query process, while VON does not have these issues. Simulation also shows that topology can be maintained quite consistently with VON [Hu et al. 2005]. We therefore choose to base this research project on the continuing studies and development of VON as it has shown to maintain consistent P2P topology in a bandwidth-efficient manner, which are crucial for true scalability in practical applications.

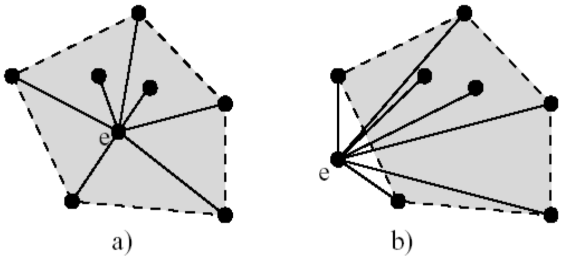

Figure 2: Design of Solipsis. Every node needs to be within a convex hull of its neighbor as in (a) and needs corrections if (b) occurs (source: [Keller and Simon 03])

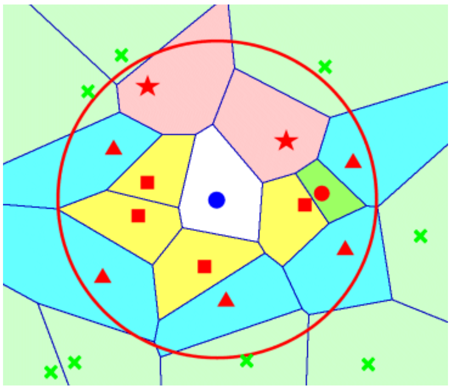

Figure 3. Large circle is the AOI boundary for the center node. Squares (▦) are *enclosing neighbors*; triangles (▲) are *boundary neighbors*; stars (★) are *both* enclosing and boundary neighbors; circle (●) represents a regular *AOI-neighbor*; crosses (⊠) represent neighbors irrelevant (i.e. outside of AOI) to the center node (source [Hu et al. 2005])

Related Work in 3D Streaming

To ensure that a 3D NVE may be as easily accessible as the WWW today, real-time streaming of 3D contents will be essential, much like how audio and video streamings have facilitated the widespread use of such contents. There are two major types of data in typical 3D contents: geometric *meshes* and graphical *textures*. Together they constitute the majority of data volume and thus will be the main focus of our discussion. Hoppe introduced the concept of *progressive meshes* [Hoppe 1996], which is a data representation method to store an arbitrary mesh as a *base mesh* and a number of progressive refinements. Rendering the 3D object requires only the base mesh, and object appearance will be progressively refined as additional data pieces are added. Progressive meshes, together with progressive representation of image data (i.e. progressive textures), form the basis of the continuous transmission, or *streaming* delivery of 3D contents over a network connection. The study of the real-time delivery of 3D data (i.e. *3D streaming*) is a fairly new field that has received attention only recently when broadband networks became available to the general public. Existing works can be roughly categorized into the streaming of a single 3D object (i.e *object streaming* [Rusinkiewicz and Levoy 2001; Fogel et al. 2001; Hosseini and Georganas 2002; Chen and Nishita 2002; Chen et al. 2003; Sahm et al. 2004]); of an entire scene with multiple objects (i.e. *scene streaming* [Hesina and Schmalstieg 1998; Teler and Lischinski 2001]); of scientific data (i.e. *visualization streaming* [Olbrich and Pralle 1999]); or of server-rendered 3D images (i.e. *image-based streaming* [Cheng et al. 2004]). Although steady progress has been made in 3D streaming, to the best of our knowledge, all existing works discuss streaming only in the context of a client-server architecture, yet the true potential in 3D streaming may need to be realized with a more scalable and distributed delivery methodology, as we will discuss later.

Related Work in P2P-based Streaming

Although audio and video streaming are matured technologies for at least 10 years, the scalability and cost issues of server-based approaches have prompted significant amount of research interests into the studies of audio and video streaming through P2P networks (e.g. CoopNet [Padmanabhan et al. 2002], layered P2P streaming [Cui and Nahrstedt 2003], Zigzag [Tran et al. 2003], oStream [Cui et al. 2004]), as P2P-based streaming can off-load server bandwidth use and may potentially shorten download time significantly (as demonstrated by file-sharing application such as BitTorrent [BT 2005]). However, 3D content downloading in an interactive scenario differs in requirements from traditional audio and video streaming, in the sense that users need not download the contents in its entirety but only those that are of current interest to the viewer. In other words, 3D streaming is as if every user is downloading a different movie to watch. At the same time, streaming of 3D contents is similar to traditional streaming in the sense that for a given 3D object, the download is sequential (as with progressive mesh and progressive texture) and prefetching techniques can be utilized.

Related Work in Distributed Storage

For a large-scale NVE to support data persistency (i.e. users status and object states will not change between login sessions), distributed data storage will become essential. The growth and popularity of user-generated contents on the Internet such as blog, file-sharing, and photo sharing also indicate that massive amount of distributed storage will be in high demand in the future Internet environment. OceanStore [Kubiatowicz et al. 2000] is one of the first projects to consider storing large amount of persistent data over a large set of distributed computer nodes. Subsequently many other distributed storage systems have been proposed (e.g. Freenet [Clarke et al. 2000], DistributedWorld [Hansen 2002] and Granary [Zheng et al. 2004]). However, comprehensive studies have yet been done on devising or finding P2P-based storage suitable for NVE purposes. The unique characteristics (e.g. user interest and data relevance is localized) and requirements (e.g. transmission latency needs to be minimized to fulfill real-time requirements) of NVE applications likely will pose different design issues to a distributed P2P storage solution.

Persistency is an essential issue in P2P-NVE. [Wang et al. 2005] propose a data persistence mechanism and implement a persistence server, called tree-structured persistence server (TSPS), to support the data management for collaborative applications. The TSPS allows states of collaborative applications to be stored in a tree fashion beside tables. By introducing TSPS, the problems of collaborative applications, such as persistence data access, asynchronous collaboration, latecomer, version control, etc., can be solved easily. [Del-Fabbro et al. 2005] presents an efficient data persistence policy for Application Service Provider architectures and avoids useless data transfers between client and servers. It also proposes “distributed storage infrastructure” to provide data transfer and storage and “request sequencing” to decrease network traffic amongst client and servers. DistributedWorld [Hansen 2002] describes the design and implementation of a persistent distributed object oriented system and programming environment. However, it is built on an existing non-distributed server called Dworkin's Generic Driver (DGD).

Research Topics

In this section, we will outline the main research topics in ASCEND as carried out in three years, explain the main problems involved, and some directions towards potential solutions. The major research topics in each year are:

First Year:	3D Streaming for P2P-NVEs
Second Year:	State Consistency Control in P2P-NVEs
Third Year:	Persistent Storage Design for P2P-NVEs

Each year will have one major research theme, and the choices of these themes are based on what we consider as the major unresolved issues regarding P2P-based NVE designs. The first year topic of 3D streaming for P2P-NVEs is based on previously published works of *Voronoi-based Overlay Network (VON)* [Hu et al. 2005], and is more or less independent from the next two year's research topics. The second year's topic of distributed state consistency will lay the foundation for the third year's study on persistent storage design. There are six major issues in the development of a NVE system, namely: *consistency, responsiveness, scalability, reliability, persistency,* and *security* [Hu 2005]. As VON has shown to possess good scalability and responsiveness [Hu et al. 2005], we intend to investigate the remaining issues of *consistency, reliability,* and *persistency* of a VON-based P2P-NVE. However, as stated earlier in our goals, although *security* is also an important issue, its solutions tend to be application-specific and less general. Security issues will thus be left as future work. The details of each years are described below:

First Year: 3D Streaming for P2P-NVEs

To allow NVE to become as easily and conveniently accessible as the WWW today, one crucial requirement is that relevant 3D contents for navigation must be obtainable in real-time. However, 3D data (i.e. geometric meshes and graphical textures) are typically large in size, which can easily reach over to hundreds of megabytes for an application, effective real-time transfer of 3D contents through *streaming* techniques thus might be needed. *3D streaming* involves two major tasks: the selection of relevant data as determined by the user's viewpoint (i.e. *object selection*), and efficient transfer of the content (i.e. *object transmission*) [Sahm et al. 2004]. For the first task, server-side visibility determination methods are generally used, while efficient representation and compression of 3D objects, as well as progressive transmission methods, are usually utilized for the second task.

The major pending issues, however, are the scalability of the system in terms of the number of concurrent users, and the amount of download time (i.e. delay) experienced by each user. As each user request will add a certain amount of processing and bandwidth burden to the server, scalability may be limited. On the other hand, a single source (i.e. the server) of 3D content also sets an upper limit to the bandwidth and latency of a given transmission.

3D streaming is inherently a processing and bandwidth intensive operation, we thus believe that for 3D streaming to be usefully adopted by large number of users, a new paradigm for data transfer is likely needed. Recent studies in using P2P to support networked virtual environments (P2P-NVE) might be utilized to support 3D streaming as well, and we plan to investigate how 3D streaming might be supported on top of *Voronoi-based Overlay Network* (VON) [Hu et al. 2005], as it has been demonstrated to possess good scalability, consistency, and reliability among existing P2P-NVE proposals.

Our initial proposal is to divide an NVE into fixed-size *cells*, each of which contains a *scene description*, which is simply summary information of the 3D objects whose *location points* (i.e. a representation point coordinate of the object) lie within the cell (Figure 4). As each node can independently calculate which cells its AOI overlaps given the dimensions of the NVE and cell size, a node's visibility may be determined without server intervention, solving the first issue of object selection. Once a node learns of which 3D objects it needs to download, it may send out requests to other user nodes within its AOI (i.e. *the AOI neighbors*), which can be learned through VON. We assume that the data of each 3D object can be decomposed into a *base piece* and many *refinement pieces*, 3D data will thus be obtained through requests to a pool of potential data sources that include the AOI neighbor and the server (as a final data source). This will solve the second issue of object transmission in a more distributed manner, as AOI neighbors likely already possess the data pieces of interest, we can effectively alleviate the server's load by sending requests to them first.

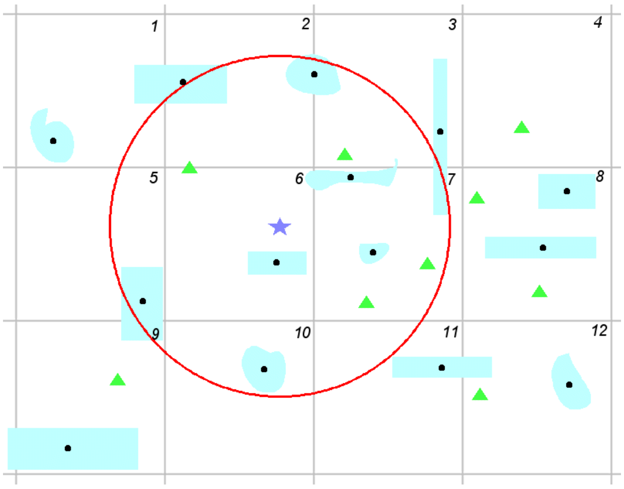

Figure 4: Schematic diagram of a NVE divided into *cells*. The big circle indicates the AOI of the center node (star) and triangles are other user nodes. Rectangular or irregular shapes are the 3D objects in the NVE, with dots indicating their *location points*. Note that cell numbers can be deterministically calculated from cell size and world dimensions by each node (source: [Hu 2005]).

As we intend to support 3D streaming by using VON, we will also need to improve VON's existing delivery mechanisms. The original proposal of VON requires all nodes to maintain direct connections and mutual visibility with one another, thus it may be called the *direct connection* (DC) model [Hu et al. 2005]. Works are also now in progress to develop a *forwarding model* (FO), where all nodes connect only to their enclosing neighbors, and receive additional messages through node relays, in order to minimize bandwidth use [Chen 2005].

While scalability has been demonstrated in the DC model, it does not take consideration or advantage of the various bandwidth capacities of different nodes. For example, nodes with more bandwidth and processing power should be able to have larger visibilities in the NVE if the users desire, or be used to help relay and process messages for the less powerful nodes. On the other hand, a pure FO model of VON will have latency not suitable for many NVE applications. A hybrid approach that combines the DC and FO models of VON in an adaptive way, in order to take full advantage of the *heterogeneity* of the P2P network thus is an important direction if practical VON-based NVE systems were to be built.

Second Year: State Consistency Control in P2P-NVEs

Although consistency has been studied extensively in parallel and distributed systems, many approaches may need to be re-adopted and re-investigated for NVE applications, due to the real-time, responsive requirements of NVE systems (i.e. message complexity and latency need to be minimized in order to allow smooth and responsive human interactions) [Zhou et al. 2002].

Consistency takes a myriad of meanings in various contexts, however, in NVE there are mainly two types of consistency to consider: state consistency and event consistency. *State consistency* refers to whether different users share the same sets of *object states* (i.e. information about an object) within the NVE (e.g. whether we both see that a glass of water is full), while *event consistency* refers to whether the users experience event occurrences in the same order. In order to synchronize object states and event ordering, messages need to be sent between each participating nodes of a NVE. However, message transmissions face the issues of *latency*, *jitter* (i.e. variance in latency) and *packet loss*, which in turn can cause *asynchrony* (i.e. event or state changes are processed at a later time than their actual occurrences due to latency), *mis-ordering* (i.e. different nodes may receive event messages in different orders due to jitter), and *mis-information* (i.e. remote nodes are not aware of the occurrence of events due to packet loss or corruption), making consistency maintenance a challenging issue.

In today's MMOG, consistency is maintained via the server by keeping an authoritative central repository of object states and a total ordering of all event messages. All clients must follow and synchronize with the server's version of states and events. This solution is simple and an authoritative server is seen by many as the natural way to maintain consistent event records. However, when adopting NVE to a P2P environment, such centralized event records may not exist, the challenge for *distributed state consistency* is therefore to avoid asynchrony, mis-ordering and mis-information in a distributed fashion.

Current view held by MMOG developers is that server-based consistency control is the only viable method. However, as consistency maintenances rely on message delivery, which in turn is strongly influenced by latency, higher latency likely will make event synchronizations at various nodes more problematic. Yet, server-based architecture necessarily require two end-to-end hops for messages to travel from the originating node to the destination nodes (via server relay), causing longer latency than if messages were to transmit directly (i.e. only one end-to-end hop) as in a P2P-based system. An interesting research topic thus is to investigate if P2P-based NVE can actually achieve better consistency than server-based approach due to smaller latency.

For state consistency, we plan to assign each node (i.e. a user's computer) in the P2P network to maintain the states of its nearby objects within the virtual environment. Whenever updates are sent to update the object states, these *object owners* would then be the authoritative node in determining the true values of object states. The reason of having object owner is to avoid concurrent updates of the same object state by two different nodes, resulting in potentially illogical operations (e.g. both users are able to drink up a glass of water). Additionally, to allow efficient load balancing in the state maintenances, it is best if ownership assignments are done dynamically.

Our initial proposal is to assume that every object state within an NVE is associated with a coordinate; we can thus designate the maintenances of its states to the node closest to it (i.e. the node whose Voronoi region contains the object's coordinate, see Figure 5). Whenever a node wishes to update the state of an object (e.g. a user wishes to close a door with his magic spell), it would send out an update request to the object's owner (i.e. the node whose Voronoi region covers the object). However, as nodes may know the coordinates of their neighbors in a slightly different manner (due to the latency in position update messages), different nodes may have different understanding as to which node is the actual owner. Update requests will thus be sent not just to the object's owner, but also to the owner's enclosing neighbors (which are the surrounding nodes sharing an edge with a particular node, indicated by the stars in Figure 5). The enclosing neighbors would then send their agree or disagree vote to the supposed object owner, and only the node that receives a majority vote will then proceed with processing the request, and propagates the result of the update to other interested nodes (i.e. nodes that could see the object).

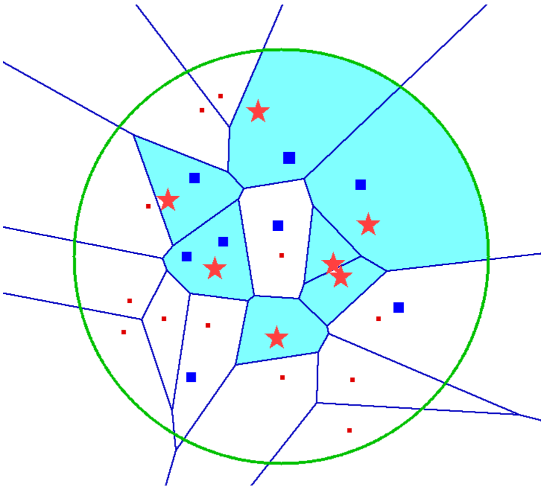

Figure 5: Object ownership determination using enclosing neighbors voting. Squares indicate objects, stars indicate enclosing neighbors (EN) of the center node.

The approach just described, however, will introduce an *ownership transfer problem*, as user nodes are constantly moving, and their nearby objects are also constantly changing. How to balance between distributing the object maintenance load and the distribution overhead (i.e. ownership transfer) efficiently and correctly thus will become one of our main research issues.

For event consistency, we plan to investigate certain distributed synchronization methods such as *bucket synchronization* [Diot and Gautier 1999] and *critical causality* [Zhou et al. 2002] to see their suitability on P2P networks and whether better approaches may be adopted. Bucket synchronization is a technique where each node in a distributed system maintains certain *time buckets* of a fixed period (e.g. 100 ms), and event messages are put into respective buckets according to their time-stamps (i.e. a mark in the logical time) once received. By queuing event messages without immediately executing them, each node will be able to preserve better event ordering according to the time stamps of event messages. Critical causality is based on the observation that not all events need to be properly ordered, but only those that are *casually related* (i.e. an event B is caused by event A). Each node thus attaches to the event message it sends out with a previously received message that triggers the local event. This way, if an event is triggered by another remote event, the receiving node will not miss the triggering event due to packet loss of the previous triggering event. We will investigate how bucket synchronization may be combined with the concept of critical causality to achieve proper event ordering in a message-efficient way.

Another issue related with consistency is the reliability of the P2P network's topology. *Packet loss* and *node failures* can potentially destroy the *topology consistency* of the P2P overlay. It is thus crucial that a P2P overlay is designed to handle such situations in order to prevent *overlay partition* [Kawahara et al. 2004]. While existing P2P-based NVE proposals have more or less some mechanisms to tolerate packet loss and node failures, no empirical studies has yet been done to compare the various proposals. On the other hand, general mechanism to handle node failures still would not prevent overlay partition when massive number of nodes fail simultaneously (which is true in other types of P2P system as well, including DHT [Stoica et al. 2001]), mechanisms to recover from overlay partition thus is also an important topic.

Potential solutions may be found from existing proposals such as checking back with a centralized server periodically (e.g. “random introduction” [Kawahara et al. 2004]). However, trade-off between the ease of centralized mechanisms and the scalability of distributed mechanisms must be carefully investigated and weighted. Another potential path of investigation is to identify “*critical nodes*” whose failures might create overlay partition. If correct and efficient mechanisms may be developed to identify and subsequently enhance the availability or reliability of these “critical nodes”, we might be able to reduce the probability of overlay partitions.

Third Year: Persistent Storage Design for P2P-NVEs

To provide a sense of realism and immersion, current MMOGs often operate 24 hours a day throughout the year except for scheduled server maintenance, so that users can login and experience the world at any time, while keeping the same user status (e.g. position in the virtual world, virtual item possessions, health and experience points, etc.). Ideally, a P2P-based NVE system should also be able to store persistent content (i.e. user stats or information about the virtual worlds) even as users come and go. However, persistent data in server-based MMOG is handled through storage in server-side databases, which would not be available in a true fully-distributed P2P environment. Persistent data storage in a P2P environment thus is challenging where issues such as data replication, migration, recovery, consistency, and security need to be considered in order to ultimately ensure *data availability* despite of constant joining and departure of nodes.

We acknowledge that there have already been extensive studies on various aspects of P2P-based storage systems [Kubiatowicz et al. 2000; Clarke et al. 2000; Hansen 2002; Zheng et al. 2004]. The approach of our study thus is to survey and adopt existing solutions that would be suitable for the unique characteristics of P2P-based NVE systems. We expect that the optimal persistent storage for P2P-based NVE and other types of P2P systems may differ as relevant content to a particular node is usually *localized* and may be found at the neighboring nodes. Consistency or security requirements may also be more relaxed for certain types of data than in generalized P2P-based storage systems. Specifically, we intend to devise solutions to support *scalable P2P storage using only a light-weight server (LWS)* for data backup, as our project goal is to provide scalable yet affordable P2P-based NVE designs where systems with various scale and budget may all be supported.

Under the assumption that a certain percentage of nodes are always online, persistency in a NVE may be achieved with enough data (i.e. object states) replications. However, as no single user node is permanently reliable, eventually we will need to rely on the server for object state maintenances. Yet, if all data updates were to be synchronized back to the server without a scheduling mechanism, the server may become overloaded. Our proposed study therefore focuses on how the server may collect updated object states from all the participating nodes in a scalable, controlled manner, while potentially utilizing client-side data aggregation and compression techniques to facilitate the message and time efficiency of such data collection.

An example of such data collection can be found in the crawling of search engines, which attempts to find updated information on large number of web servers through scheduled visits of each site. Bandwidth efficiency and the discovery latency however are the main problems.

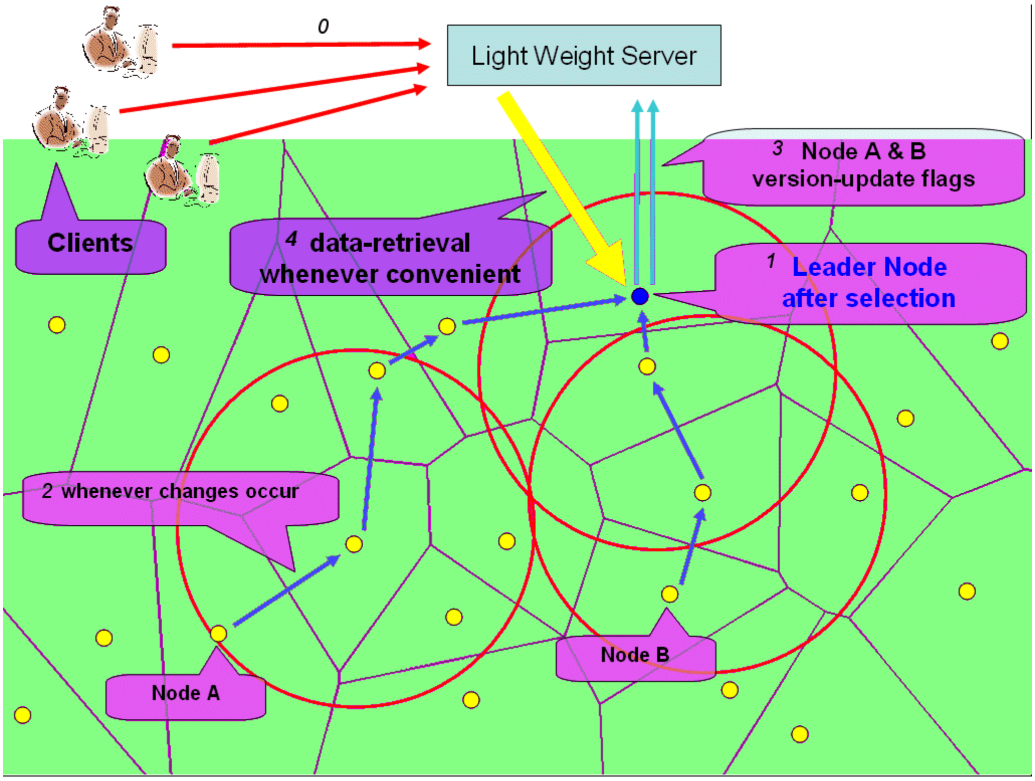
0
Light Weight Server
3 Node A & B version-update flags
Clients
4 data-retrieval whenever convenient
1 Leader Node after selection
2 whenever changes occur
Node B
Node A

Figure 6: Design for a light-weight server (LWS)-based persistent storage. (Step 0) Clients connect to a LWS first to discover an initial set of AOI neighbors (Step 1) Leader node is elected locally (Step 2) Whenever states are updated, such as Node A or Node B in the figure, updates are propagated to the leader node. (Step 3) The leader node notifies the server that new states are ready for backup. (Step 4) The server retrieves aggregated and compressed updates from the leader node according to its schedule.

Our initial proposal involves the election of *leader nodes,* which are responsible to collect state updates from neighboring nodes and synchronize the updates with the server. Meanwhile, any state update will trigger a version-update flag to the light weight server to notify for scheduling data-retrieval at a later time convenient to the server. In our design, a light weight server is assumed to be reliable and persistent, and acts as the first contact point of entering into the P2P-NVE, as well as the initial source for all object states (Figure 6).

Some potential problems to be addressed are the selection of leader nodes, as well as up-to-date data aggregation from peers, and we discuss certain existing approaches which we might adopt below:

Leader nodes election

Leader nodes are important elements in our approach, since they provide the ability for light-weight servers to retrieve the most up-to-date data. However, leader nodes are prone to failure if excessive bandwidth is consumed. Our concept of leader nodes is similar to [Su and Lee 2004], who proposes a gateway nodes recovery approach, which is fully distributed and allows for surviving gateway nodes to recover other gateway nodes that have failed. [Basagni et al. 2004] proposes the generation of a *clusterhead* via the DCA algorithm, which is based on a generic "weight" attribute associated to each node. Each Wireless Sensor Network (WSN) node in LEACH [Handy et al. 2002.] rotates to be cluster-head to ensure fairness. In the same way, we could rotate nodes in a VON-based NVE as leader nodes. The popular peer-to-peer file sharing system Kazaa [Kazaa] implements *supernodes* that have strong processing and bandwidth capacities. Regular peers connect to a certain supernodes which act as their local servers. [Lo et al. 2005] define the *supernode selection problem*, which has emerged across a variety of P2P applications. Supernode selection involves the selection of a subset of the peers to serve a special role as leaders. The supernodes must be well-dispersed throughout the P2P overlay network to balance the load.

Data Aggregation

Aggregation has been discussed in different contexts such as large scale databases [Gray et al. 1997], active networks [Raz and Shavitt 2002], and wireless sensor network applications [Intanagonwiwat et al. 2000; Madden, Szewczyk et al. 2002; Madden, Franklin et al. 2002]. In network aggregation schemes such as Astrolabe [Renesse et al. 2003] and TAG [Madden, Franklin et al. 2002], hierarchical aggregation in the network is performed to reduce bandwidth consumption and achieve better load-balancing. TAG simplifies aggregation without the SQL clauses *Group By* or *Having* while Astrolabe aggregates data in large distributed systems hierarchically. A related work also based on a hierarchical approach is [Gupta et al. 2001]. While building hierarchies indeed reduces the cost of finding the aggregates, it introduces additional overhead in maintaining this hierarchical topology within a dynamic and distributed environment. Moreover, due to the hierarchical organization, extra efforts and protocols are needed to broadcast the result continuously for all nodes to know. In gossip-based aggregation [Montresor et al. 2004], the state of a node is a numeric value. In a practical setting, this value can be any attribute of the environment, such as the load or the storage capacity. The task of the protocol is to calculate an aggregate value over the set of all numbers stored at nodes. As long as the overlay network remains connected, link failures do not affect the final value, but only slow down the aggregation process. [Li et al. 2005] design a algorithm for aggregation over existing DHTs. It builds an aggregation tree in a bottom-up manner by mapping nodes to their parents in the tree with a predefined parent function. [Müller 2005] proposes a process to aggregate player's information to a two-dimensional histogram for MMOG. The aggregation performs on a dynamic Delaunay triangulation network with compass routing in a grouped NVE.

Another goal in the third year is to integrate the research results of P2P-based 3D streaming, consistency control and persistent storage into a coherent set of interrelated programming libraries so that P2P-based NVE applications may be developed and supported with minimal redundant efforts. We will apply these novel techniques to build a NVE environment in collaboration with other researchers in the fields of learning and education, so that practical issues in the development and deployment of P2P-based NVE systems may be discovered and studied.

In summary, each of the three years will have major and minor research issues, which are listed below:

Year	***Major Topic***	***Minor Topic***
First	3D Streaming on P2P Network	Hybrid Model of VON
Second	Distributed State Consistency	Reliability in P2P Topology Consistency
Third	Persistent Data Backup on P2P	Integrations and Applications

Expected Results

The complete execution of the ASCEND project will bring three research contributions: 1) a validated 3D streaming mechanism using P2P networks 2) a consistency control mechanism that ensures both the topology and event consistency in a P2P environment 3) a light-weight server-based P2P storage system designed for NVE applications. Specific milestones for each year are described below:

First Year

In the first year, we expect to have done thorough surveys in related fields and created three main proposals regarding the adoption of 3D streaming on P2P networks. The milestones are:

- Parallel streaming mechanisms of 3D contents (meshes and textures) on a P2P network
- A fully-distributed visibility determination method
- An enhanced VON that combines direct and relayed message transmissions

Second Year

The second year will mainly focus on the state and event consistency control for P2P-based NVE. Evaluation and validation of proposed solutions will also be carried out. The milestones are:

- Distributed consistency control mechanisms that ensure proper event ordering
- Ownership transfer mechanisms for object states in P2P environment
- Overlay partition recovery mechanism and reliability simulations of VON

Third Year

The goal of the third year is a distributed P2P data storage for NVE applications utilizing only a light-weight server, and the application of the research results in the first two years into an actual P2P-based NVE application, likely for education and e-learning purposes. Most of the components in the ASCEND library will be ready, which include components for 3D streaming, consistency control, and persistent storage support. Efforts will also be spent to transfer the implementation to the open source community for the continuing development and maintenances of the library. The milestones are:

- Scalable and distributed backup techniques utilizing only a light-weight server.
- Data availability mechanism exploiting the locality of user interests
- A prototype P2P-based NVE system for educational purpose

P2P-based network virtual environment is an emerging field that likely will become an important application domain for P2P systems, as indicated by the emergence of MMOG, P2P file-sharing, and hardware trends in CPU, broadband, and consumer-level 3D acceleration. The ASCEND project seeks to take on a leadership position in this exciting new research by thoroughly investigating selected major aspects related to fundamental issues in P2P-based NVE systems.

As the output of ASCEND project will include a set of related programming libraries to support the development of P2P-based NVE systems (i.e. the ASCEND Library), throughout the three years we will also be hosting research results on the open source website SourceForge [ASCEND 2005] and engaging in community building so that the ASCEND project can continue its developments after this three year period.

Implications of the ASCEND project will be three-fold:

1) Increase in our understandings of how to construct practical P2P-based NVE systems with proposed solutions in related issues such as real-time 3D data delivery, persistent storage, and consistency control in a distributed P2P environment.

2) Creation of a set of foundational open-source programming library (i.e. the ASCEND library) which will facilitate further the research, development, and adoption of P2P-based NVE systems for new domain areas such as education, training, and e-commerce.

3) The training and development of a core team of researchers and developers who will have in-depth knowledge on various aspects of P2P-based NVE issues, as well as practical experience in the development and integration of related programming libraries. This will hopefully form the basis of a new open source based software industry specialized in the development of 3D-based virtual environments for various purposes.

We believe that the time is approaching for a viable interactive *medium* to emerge on the Internet that is 3D-based, highly interactive, and user-empowering. Within a few years, adequate hardware infrastructure will mature to support such NVE systems on a large-scale, it is thus essential, even necessary, to start off research efforts now to be ahead in this important Internet application domain.

References

[ASCEND 2005] ASCEND Project Home Page. http://ascend.sourceforge.net

[Basagni et al. 2004] S Basagni, M Mastrogiovanni, C Petrioli, "A performance comparison of protocols for clustering and backbone formation in large scale ad hoc networks" Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference

[Baset and Schulzrinn 2004] Salman A. Baset and Henning Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol," Columbia University Tech. Rep. CUCS-039-04, 2004.

[BT 2005] Bittorrent Homepage http://www.bittorrent.com

[Chen and Nishia 2002] Bing-Yu Chen and Tomoyuki Nishita, "Multiresolution streaming mesh with shape preserving and qos-like controlling," in Proc. ACM Web3D, 2002, 35–42.

[Chen et al. 2005] Tsu-Han Chen, Jui-Fa Chen and Shun-Yun Hu, "A Forwarding Model for Voronoi based Overlay Network", VAST Technical Report. VAST-2005-01. Available at: http://vast.sourceforge.net/docs/pub/VAST-TR-2005-01.pdf

[Chen et al. 2003] Zhihua Chen, Bobby Bodenheimer, and J. Fritz Barnes, "Robust transmission of 3d geometry over lossy networks," in Proceeding of the eighth international conference on 3D Web technology, 2003, 161–ff.

[Cheng et al. 2004] Liang Cheng, Anusheel Bhushan, Renato Pajarola, and Magda El Zarki, "Real-time 3d graphics streaming using mpeg-4," in Proceedings IEEE/ACM Workshop on Broadband Wireless Services and Applications, 2004.

[Clarke et al. 2000] Ian Clarke, Freenet Homepage http://freenet.sourceforge.net/2000

[Croquet 2005] Croquet project Home Page. http://www.opencroquet.org/

[Cui and Nahrstedt 2003] Yi Cui, Klara Nahrstedt, "Layered Peer-to-Peer Streaming" Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video June 2003

[Cui et al. 2004] Y Cui, B Li, K Nahrstedt, "oStream: Asynchronous Streaming Multicast in Application-Layer Overlay Networks" IEEE Journal on Selected Areas in Communications, 2004

[Diot and Gautier 1999] Christophe Diot and Laurent Gautier, "A distributed architecture for multiplayer interactive applications on the Internet," IEEE Network, vol. 13, no. 4, pp. 6-15, 1999.

[Fogel et al. 2001] Efi Fogel, Daniel Cohen-Or, Revital Ironi, and Tali Zvi, "A web architecture for progressive delivery of 3d content," in Proc.ACM Web3D, 2001, 35–41.

[Gray et al. 1997] Jim Gray,Surajit Chaudhuri,Adam Bosworth,Andrew Layman,Don Reichart,Murali Venkatrao, Frank Pellowand Hamid Pirahesh, "Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-Tab, and Sub Totals" Data Mining and Knowledge Discovery, 1997

[Gupta et al. 2001] I Gupta, R Renesse, KP Birman, "Scalable Fault-Tolerant Aggregation in Large Process Groups" DSN, 2001

[Handy et al. 2002] MJ Handy, M Haase, D Timmermann, "Low energy adaptive clustering hierarchy with deterministic cluster-head selection" IEEE MWCN, 2002

[Hansen 2002] Hansen, "A Distributed Persistent World Server using Dworkin's Generic Driver" Cand. Scient. Thesis 2002

[Hesina and Schmalstieg 1998] Gerd Hesina and Dieter Schmalstieg, "A network architecture for remote rendering," in Proc. Second International Workshop on Distributed Interactive Simulation and Real-Time Applications 88, 1998.

[Hosseini and Georganas 2002] Mojtaba Hosseini and Nicolas D. Georganas, "Mpeg-4 bifs streaming of large virtual environments and their animation on the web," in Proc. ACM Web3D, 2002, 19–25.

[Hoppe 1996] Hugues Hoppe, "Progressive meshes," Computer Graphics (SIGGRAPH '96 Proceedings) (1996), 99–108.

[Hu et al. 2005] Shun-Yun Hu, Jui-Fa Chen, and Tsu-Han Chen, "VON: A scalable peer-to-peer network for virtual environments," accepted publication to IEEE Network, 2005, http://vast.sourceforge.net/docs/pub/2005VON.pdf

[Hu and Liao 2004] Shun-Yun Hu and Guan-Ming Liao, "Scalable peer-to-peer networked virtual environment," in Proc. ACM SIGCOMM 2004 Wksp. on NetGames '04, Aug. 2004, pp. 129-133.

[Hu 2005] Shun-Yun Hu, "Scalable peer-to-peer networked virtual environment," Master's thesis, Tamkang Univ., Taiwan, 2005. http://vast.sourceforge.net/docs/pub/2005_hu_master_thesis.pdf

[Zheng et al. 2004] J Hu, M Li, W Zheng, "Granary: Architecture of Object Oriented Internet Storage Service" Proceedings of the E-Commerce Technology 2004

[IGDA 2004] IGDA, "2004 Persistent Worlds Whitepaper," http://www.igda.org/online/IGDA_PSW_Whitepaper_2004.pdf

[Intanagonwiwat et al. 2000] Intanagonwiwat et al., "Directed diffusion: A scalable and robust communication paradigm for sensor networks" 6 th Annual International Conference on Mobile Computing

[Iyer et al. 2002] Sitaram Iyer, Anthony Rowstron, and Peter Druschel, "Squirrel: A decentralized peer-to-peer web cache," in Proc. 21st ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC), 2002.

[Kazaa] Kazaa Homepage http://www.kazaa.com

[Kawahara et al. 2004] Yoshihiro Kawahara, Tomonori Aoyama, and Hiroyuki Morikawa, "A peer-to-peer message exchange scheme for large-scale networked virtual environments," Telecomm. Sys., vol. 25, no. 3-4, pp. 353–370, 2004.

[Keller and Simon 2003] Joaquin Keller and Gwendal Simon, "Solipsis: A massively multi-participant virtual world," in Proc. Int. Conf. Parallel and Dist. Tech. and App. (PDPTA 03), 2003, pp. 262-268, 2003.

[Knutsson et al. 2004] Bjorn Knutsson, Honghui Lu, Wei Xu, and Bryan Hopkins, "Peer-to-peer support for massively multiplayer games," in Proc. IEEE INFOCOM, Mar. 2004, pp. 96-107.

[Kubiatowicz et al. 2000] John Kubiatowicz, David Bindel, Yan Chen, Steven Czerwinski, Patrick Eaton, Dennis Geels, Ramakrishna Gummadi, Sean Rhea, Hakim Weatherspoon, Westley Weimer, Chris Wells, and Ben Zhao, "OceanStore: An architecture for global-scale persistent storage," in Proceedings of Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX), ACM Press, New York, 2000.

[Lineage 2005] Lineage Home Page. http://en.wikipedia.org/wiki/Lineage_(game)

[Madden, Szewczyk et al. 2002] S Madden, R Szewczyk, MJ Franklin, D Culler, "Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks" WMCSA, 2002

[Madden, Franklin et al. 2002] S Madden, MJ Franklin, JM Hellerstein, W Hong, "TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks" OSDI, 2002

[Miller and Thorpe 1995] DUNCAN C. MILLER and JACK A. THORPE, "SIMNET: The Advent of Simulator Networking," in Proc. IEEE, vol. 83, no 8, pp. 1114-1123, Aug. 1995.

[MS 2005] Microsoft windows Vista Home Page. http://www.microsoft.com/windowsvista/default.mspx

[Montresor et al. 2004] A Montresor, M Jelasity, O Babaoglu, "Robust aggregation protocols for large-scale overlay networks" Dependable Systems and Networks, 2004 International Conference 2004

[Li et al. 2005] J Li, K Sollins, DY Lim, "Implementing Aggregation and Broadcast over Distributed Hash Tables" ACM SIGCOMM Computer Communication Review, 2005

[Lo et al. 2005] V Lo, D Zhou, Y Liu, C GauthierDickey, J Li, "Scalable Supernode Selection in Peer-to-Peer Overlay Networks" Hot Topics in Peer-to-Peer Systems, 2005

[Olbrich and Pralle 1999] Stephan Olbrich and Helmut Pralle, "Virtual reality movies – real-time streaming of 3D objects," Computer Networks (Amsterdam, Netherlands: 1999) 31, 21, 2215–2225, 1999.

[Padmanabhan et al. 1995] Venkata N. Padmanabhan, Helen J. Wang, Philip A. Chou, "Distributing Streaming Media Content Using Cooperative Networking" SB Wicker - 1995 - Englewood Cliffs, NJ: Prentice Hall

[Raz et al. 2002] D Raz, Y Shavitt, "New models and algorithms for programmable networks" Computer Networks, 2002

[Renesse et al. 2003] R van Renesse, KP Birman, W Vogels, "Astrolabe: A Robust and Scalable Technology for Distributed System Monitoring, Management, and Data Mining" ACM Transactions on Computer Systems, 2003

[Rusinkiewicz and Levoy 2001] Szymon Rusinkiewicz and Marc Levoy, "Streaming QSplat: A viewer for networked visualization of large, dense models," in Proc. Symp. Interactive 3D Graphics, 63–69, 2001.

[Sahm et al. 2004] Jorg Sahm, Ingo Soetebier, and Horst Birthelmer, "Efficient representation and streaming of 3d scenes," Computers & Graphics 28, 1, 15–24, 2004.

[Singhal and Zyda 1999] Sandeep Singhal and Michael Zyda, Networked Virtual Environments: Design and Implementation, ACM Press, New York, 1999.

[Stoica et al. 2003] Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for Internet applications," IEEE/ACM Trans. Networking, vol. 11, no. 1, pp. 17-32, 2003.

[Teler and Lischinski 2001] Eyal Teler and Dani Lischinski, "Streaming of complex 3d scenes for remote walkthroughs," in Proc. EUROGRAPHICS 20, 3, 2001.

[Tran et al. 2003] DA Tran, KA Hua, T Do, "ZIGZAG: An efficient Peer-to-Peer Scheme for Media Streaming" Proc. of IEEE INFOCOM, 2003

[VAST 2005] VAST Project Home Page. http://vast.sourceforge.net

[Wang et al. 2005] CM Wang, HM Chen, GC Lee, SF Hong, "A Tree-Structured Persistence Server for Data Management of Collaborative Applications" Proceedings of the 19th International Conference on Advanced Information Networking and Applications - Volume 2 AINA '05

[Su et al. 2004] WW Su, SJ Lee, "An Adaptive and Fault-Tolerant Scheme For Gateway Assignment In Sensor Networks" Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference

[WoW 2005] World of Warcraft Home Page. http://www.worldofwarcraft.com/

[Zhou et al. 2002] Suiping Zhou, Wentong Cai, Stephen J. Turner, and Bu-Sung Lee. "Critical Causality in Distributed Virtual Environments", in Procs. of 16th IEEE/ACM/SCS Workshop on Parallel & Distributed Simulation (PADS 2002), pp.53-59, Washington DC, USA, May 2002.