國 立 中 央 大 學

資 訊 工 程 研 究 所

碩 士 論 文

Discovery of Physical Neighbor for P2P Networked Virtual Environment

網路虛擬環境與點對點傳輸架構下之實體鄰近節點選擇

研 究 生：吳 昌 樺

指導教授：曾 黎 明 博士

中 華 民 國 九 十 七 年 十 月

國立中央大學圖書館
碩博士論文電子檔授權書

(95 年 7 月最新修正版)

本授權書所授權之論文全文電子檔(不包含紙本、詳備註 1 說明)，為本人於國立中央大學，撰寫之碩/博士學位論文。(以下請擇一勾選)

( )同意 (立即開放)
(✓)同意 (一年後開放)，原因是：＿＿尚未投稿＿＿
( )同意 (二年後開放)，原因是：＿＿＿＿＿＿＿＿
( )不同意，原因是：＿＿＿＿＿＿＿＿

以非專屬、無償授權國立中央大學圖書館與國家圖書館，基於推動「資源共享、互惠合作」之理念，於回饋社會與學術研究之目的，得不限地域、時間與次數，以紙本、微縮、光碟及其它各種方法將上列論文收錄、重製、公開陳列、與發行，或再授權他人以各種方法重製與利用，並得將數位化之上列論文與論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

研究生簽名: ＿＿吳昌樺＿＿ 學號：＿＿955302030＿＿

論文名稱:＿＿網路虛擬環境與點對點傳輸架構下之實體鄰近節點選擇＿＿

指導教授姓名：＿＿曾黎明＿＿

系所 ：＿＿資訊工程＿＿所 ☐博士班 ☑碩士班

日期：民國＿97＿年＿10＿月＿7＿日

備註：

1. 本授權書之授權範圍僅限電子檔，紙本論文部分依著作權法第 15 條第 3 款之規定，採推定原則即預設同意圖書館得公開上架閱覽，如您有申請專利或投稿等考量，不同意紙本上架陳列，須另行加填聲明書，詳細說明與紙本聲明書請至 http://blog.lib.ncu.edu.tw/plog/ 碩博士論文專區查閱下載。
2. 本授權書請填寫並**親筆**簽名後，裝訂於各紙本論文封面後之次頁（全文電子檔內之授權書簽名，可用電腦打字代替）。
3. 請加印一份單張之授權書，填寫並親筆簽名後，於辦理離校時交圖書館（以統一代轉寄給國家圖書館）。
4. 讀者基於個人非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

摘要

網路虛擬實境（NVE)，是網路互動技術的一種，也是一個相關的新領域，結合了虛擬實境跟網際網路的特點，提供給人們一種身歷其境的體驗。隨著愈來愈多的人們進入這個世界，其中，快速成長跟大眾化的流行，帶來對網路虛擬環境中大規模化的要求，這將會對現今的網路跟用戶端伺服器架構帶來很大的挑戰。

點對點傳輸架構的優點是每多一台機器進入,就可以多帶來一分多的資源可用。在網路虛擬實境中，這個優點非常適合取代當要求愈多，伺服器的負載就會愈多的早期用戶端伺服器架構。因此，有愈來愈多具延展性，有效率而且簡單的使用點對點傳輸架構方法被提出，來建構網路虛擬實境。透過與其他節點的組織了解跟合作，在鄰近節點找到之後，就可以實現在點對點傳輸架構的網路虛擬實境（P2P-NVE）下，其系統架構的延展性跟3D物件傳遞的方便性。

然而，隨意的在虛擬世界挑選鄰近節點，卻沒有考慮到任何有關真實世界的資訊的話，將有可能會因此選擇到結果比較不好的節點，例如:節點間的網路延遲過長。所以根據這點，我們提出了一種把在真實世界裡，比較接近的節點分在同一群，並且透過節點間的分工合作，來維持其結構完整的方法（Discovery of Physical Neighbor: DPN)；然後結合虛擬跟真實的考量，試著將網路延遲過長的副作用減到最低。最後，我們也驗證了這個實體鄰近節點方法的效能，的確能對點對點傳輸架構下的網路虛擬實境有所幫助。

關鍵字：網路虛擬實境，點對點傳輸網路，鄰近節點，地標點。

Abstract

Networked virtual environment (NVE), an Internet interactive application technology, combined virtual reality with Internet and providing immersion experience to the people is a relatively new field. Because of the more and more people involving in this fantasy world, the rapid growth and popularity of large-scale NVE will induce serious challenges to existing network and client-server architecture.

Bringing the additional resources with each peer joined is the advantage of the P2P architecture. It is very suitable to take the responsibility of the client-server architecture in the NVE. In this point of view, there come many scalable, efficient, and simple methods in P2P-NVE establishment. By organizing and realizing the neighbor discovery through mutual node collaborations, the P2P-NVE can remains its scalability and obtains the 3D content conveniently.

However, the peer randomly chooses the logical neighbor in virtual world without more information about physical distance may cause a serious latency in the P2P-NVE. According to this assumption, we propose a method called Discovery of Physical Neighbor (DPN) by grouping the peer with the physical short distance to ease the negative influence off. And put the maintenance of this system model to the peer cooperation. Finally, the performance of the approximate physical neighbor discovery is obviously testified in the P2P-NVE.

Keywords: networked virtual environment, peer-to-peer network, P2P-NVE, neighbor, landmark.

Acknowledgements

How time does fly. Yesterday I was just a rookie of this big family and today I am going to graduate from NCU. It truly is amazing how quickly time goes. Everything became hazily little by little. It is gone with the wind but lie down memories in my heart only. These priceless moments will never be forgotten.

I first deeply appreciate Professor Li-Ming Tseng, the remarkable and superior mentor, for his attentive direction. He pilots me to the sanctuary of research and inspires the emotion of pursuing knowledge in my life. And I also wish to express my gratitude to Professor Li-Der Chou and Professor Yi-Ming Chen for their useful guidance for correcting my thesis. Moreover, my appreciation goes to Professor Ce-Kuen Shieh and Professor Chu-Sing Yang for reading my manuscript and gave me the helpful suggestions.

Second, I would like to thank Hung-Chang Yang, Shun-Yun Hu and other companions in the Distributed System Laboratory. We share the joys and tears to each other in these years during my graduate livelihood. The great appreciations for full supports of department managers R.G. Liu and H.T. Lin, sector manager C.W. Lai, supervisors, and colleagues in tsmc are gushed from my heart.

Finally, I felt an immense gratitude to my father, mother, brother and those close friends for their kindly support. Nothing recedes like success. I will keep studying, learning, and working hard in my future.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

By the increasing network bandwidth, the request of different kinds of Internet function becomes more and more booming in this world now. Networked virtual environment (NVE) [1] [2], an Internet interactive application technology, combined virtual reality with Internet and providing immersion experience to the people is a relatively new field. The NVE, known as distributed virtual environments, is developed with military simulators [3] in the 1980s and has evolved to Massively Multiplayer Online Games (MMOG) [4] in the mid-1990s. Now and days, MMOG is the most popular domain of NVE and grows to a virtual, dream, and fantasy world where multiple participants interacting with each other.

Today, in order to build the proximally realistic NVE, most MMOGs need to create a lot of 3D contents (e.g. World of Warcraft, a famous MMOG, is over 5 GB). Existing MMOG companies, however, choose the client-server architecture for 3D content delivery. Because the 3D contents are large size and process-intensive, server-side network bandwidth and computing resource will be in the heavy loading when serving a lot of players at the same time. Server is the bottleneck if all data download using client-server architecture. Therefore, the alternative way for the game users to obtain the 3D contents is through pre-installation CDs or by prior downloads. According to the growth entertainment industry, a quicker, better, and simpler manner for huge 3D contents delivery technology becomes a regarding issue.

Besides the above discussed topic, the scalability is also an important issue if planning to build truly massive worlds and applications, which thousands upon thousands of people can enjoy. The rapid growth and popularity of large-scale NVE will induce serious challenges to the existing network and client-server architecture. Peer-to-peer (P2P) architecture, a new class of technology, appears and be took to solve the scalability problem of large-scale NVE.

P2P architecture is different from client-server architecture in that the basic units are commodity PCs with equivalent functionality. In P2P architecture, each peer is a server role and a client role respectively. Most P2P architectures are distributed system without any centralized control or hierarchical organization. In the application of P2P architecture, it can guide to the three directions [5]. They are distributed file sharing, person-to-person messaging, and distributed computing systems. Nowadays, because the P2P systems are composed of commodity hardware and promise scalability and affordability by contributing user resources to maintain the system, it is undoubted a flourishing technology in the field of network.

Because of above advantages, it is scale enough to apply in the NVE by adopting P2P technology. The occurring combined approach is called P2P-NVE and has a lot of successful researches today. There are great achievements of neighbor discovery and content delivery by P2P architecture in the virtual world. However, the new problems go along with the growth of P2P-NVE. One problem is that if the peers or neighbors are random and logical selected by each other but without more information about physical distance may cause a serious latency in the P2P-NVE.

Considering the characteristic of P2P-NVE, we will propose a method called Discovery of Physical Neighbor (DPN) in this thesis. In our method, DPN, we separate the peers into sever group by the close physical distance first. And then maintain the group with tree architecture by peers to alleviate the loading of the content servers. Moreover, to get the content as soon as possible, we institute the priority of content delivery to complete our main purpose. Finally, DPN is theoretical proved by the results of related simulation.

The rest of the thesis is organized as follows: Section II discusses related work, and our method for P2P-NVE issue is presented in Section III. To evaluate the performance of our method, Section VI describes the results of related simulations. Finally, conclusions are given in Section V.

Chapter 2

Related Work

2.1 P2P Networked Virtual Environment

NVE is artificial world where each user interacts with other human or computer player and acts as a virtual identity in the Internet. In NVE, everything is similar to the real world. Participants may perform different actions such as moving to new locations, looking around at the surroundings, using items to do something, etc. Today, MMOG is an important and arguably the most successful subset of NVE. The client-server architecture for game content delivery is adopted by MMOG to provide an entertaining experience to people. Facing the more and more people joining to this world day by day, it focuses MMOG more on continuous supplies of game contents to people. However, resources in any given system are usually finite and are insufficient when new nodes are continuing added.

Thus, in the client-server architecture, as long as resources can be increased, the framework is maintainable [6]. Server resource-growing and decentralized end-point resource consumption are two properties to counter resources depletion [7]. The first idea comes out is dealt with by provisioning more hardware [8] [9]. Consequentially, many previous researches have been taken in the approaches of server resource extension. Using multiple servers, server-cluster or large amount of middleware to provide plenty resources request has once became a popular solution, especially for commercial NVE [1] [10]. However, server-centered approach is a costly way for server-side bandwidth, hardware, and maintenance, which limits the numbers of potential NVE development.

From the above impression, we expect that whether system resources do not deplete while new users join (Figure2-1 (a)). And it also consists of a wide range of concurrent users without compromising certain quality of services. In other words, the system can accommodate the increasing new users within immobile resources provided by servers. The issue in this article is so called Scalability and one of the characteristics (Consistency, Responsiveness, Scalability, Persistency, Reliability, and Security) in building a success NVE system [11]. Therefore, in order to build a truly scalable NVE, we need architecture that can grow its resource, and does not require centralized resource when additional users join. A P2P architecture may alleviate this issue due to that the resources are brought into the system with each additional node attached (Figure2-1 (b)). To treat this unique requirement of NVE, we argue that P2P-NVE is worthy of study and poses as a new class of NVE system.

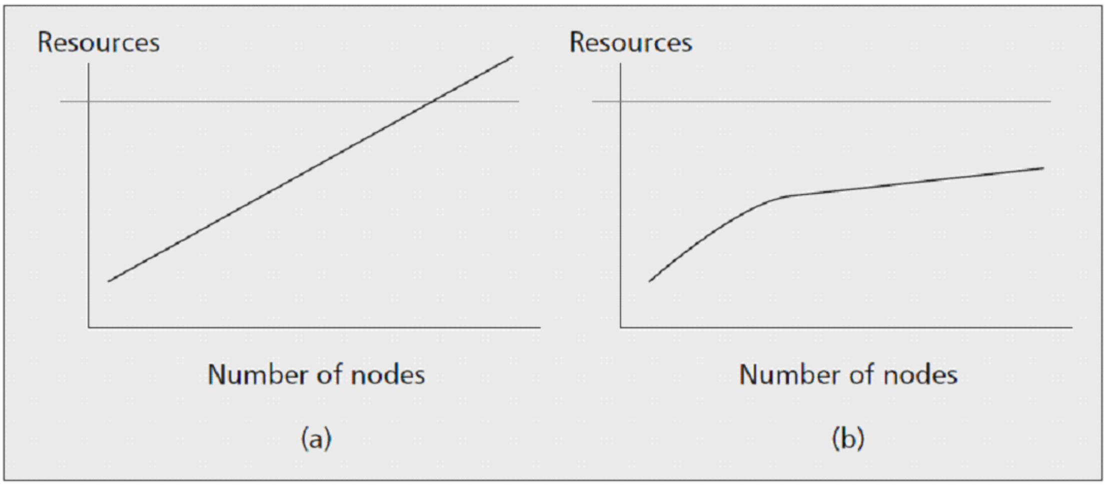


Figure 2-1：Scalability analysis (a)client-server (b)peer to peer [11]

The first step is how to gain the resources from centralization to each node. In the real world, although many users and events may exist plenty, each user is only affected by nearby users or events. Thus, we can convert this actual condition into a fundamental concept called area of interest (AOI) in a generalized NVE. Users can be conceptually seen as coordinate points on a 2D plane (as nodes), and each user with a limitary visibility range. As showed in Figure 2-2, we capture a snapshot from Google map and mark several red circles on it to denote AOI. Because each node's AOI is limited, the desired content is localized and easily identified as the messages generated by other nodes within the AOI. And it may be stated as to given a node's position and AOI, find all neighboring nodes within the AOI. That is if the structure can be established using P2P in the NVE, the scalability can be held. The central resource search in P2P-NVE systems thus becomes a neighbor discovery problem. The following discussions are some recent P2P-NVE proposals.

Figure 2-2：Area of Interest (AOI)

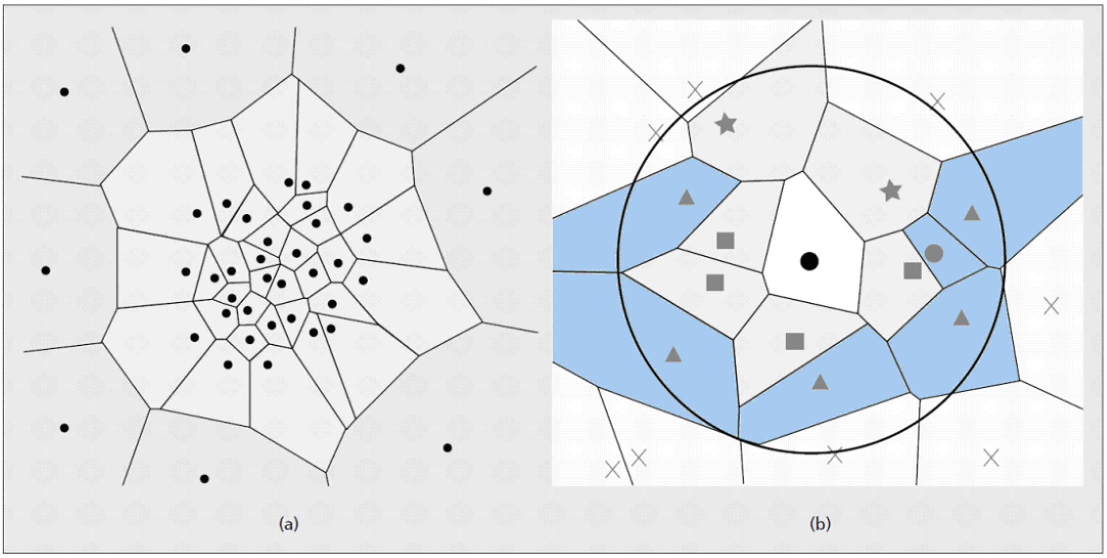
Enhanced Point-to-Point [12] allows transmission only under mutual visibility and update-free regions (UFRs) define pairs of mutually invisible regions in the NVE. Frontier sets [13] provide an improved approach for UFRs. However, UFRs do not scale, as each node needs to negotiate with all other nodes and excessive message exchanges can happen if there are too many neighbors. SimMud [14] uses Scribe, an application-layer multicast built on the DHT Pastry, and divides the NVE into some fixed-size regions and managed by a promoted super-node. The weak point is that a super-node can be a bottleneck when crowding in the region is happened. Otherwise, message latency may be up to several seconds due to relays. To reduce latency, Zoned Federation [15] uses DHT only for topology connectivity, while regular nodes connect directly to the region's super-node. It is too pity that crowding can still be a concern.

Kawahara et al. describes a fully-distributed scheme, Neighbor-List Exchange [16], in which each node directly connects with a fixed number of nearest neighbors and constantly exchanges neighbor lists in order to discover new nodes. Although direct connections minimize latency, constant list exchanges incur transmission overheads. When separated by large distances, nodes may also lose mutual contacts and cause overlay partitions. To ensure global connectivity, the Message Interchange Protocol (MIP) [17] keeps at least one neighbor in each of the four cornering directions of a node. However, the frequency of list exchange is still a delicate trade-off between neighbor discovery timeliness and bandwidth use. Therefore, a better approach is to notify new neighbors only when necessary. Solipsis [18] is another fully-distributed system, where each node connects to all the neighbors within its AOI. Neighboring nodes serve as the "watchmen" for approaching foreign nodes and neighbor discovery is achieved by notifications from known neighbors. Nevertheless, neighbor discovery is occasionally incomplete when incoming nodes may be unknown to directly connected neighbors. In other cases, active queries are required when the within-convex-hull property is violated, which slows down neighbor discovery [19].

Federated peer-to-peer [20] is a kind of client-server hybrid architecture. Nodes organize into various groups and are managed by provisioned hardware called multicast reflectors. Strictly speaking, client-server hybrids are server-cluster variants where the combined bandwidth of specialized servers determines the scalability limit. Contrary to client-server hybrid, MOPAR [21] is a P2P hybrid architecture and an interesting mix of DHT, neighbor-list exchange, and direct transfer. The NVE is divided into hexagonal cells, such as NPSNET, and DHT is used to ensure global connectivity. Within each cell, a master node maintains a list of slave nodes and exchanges the list periodically with neighboring masters. Slave nodes are notified of new neighbors by the masters. To avoid latency, slave nodes exchange messages directly among themselves. However, the selection of adequate masters is a concern, especially when crowding is considered. It has still a familiar issue in this and other neighbor-list exchange schemes. The frequency of slave-node list exchange is also a difficult trade-off between bandwidth use and neighbor discovery timeliness.

Various P2P-NVE schemes differ in their degree of decentralization. In general, the more distributing the schemes design, the more responsive the system becomes. These P2P designs are potentially more scalable than client-server architecture. However, there are still the most common issues occurring at the expense of increased bandwidth utilization, correct and timely neighbor discovery, and user crowding tolerance. Consequentially, the new proposal called Voronoi-based Overlay Network (VON) [11] is presented in P2P-NVE.

It is the practice of Voronoi diagrams [22] to solve the neighbor discovery problem in a fully-distributed, bandwidth efficient, and low-latency manner. Given a number of points (called nodes) on a 2D plane, a Voronoi diagram partitions the plane into the same number of Voronoi regions, such that each region contains all the points closer to the region's site than to any other site (Figure 2-3 (a)). The Voronoi diagram defines a spatial relationship between various nodes and may be used to find the k-nearest neighbor of any site efficiently. Since Voronoi diagrams have been widely studied and applied to diverse fields, such as computational geometry and mobile computing, it do not need to consider the specifics of Voronoi construction but assume that good algorithms exist. In this proposal, each node in VON is represented as a site in the Voronoi diagram. And then define AOI neighbors as the nodes whose positions are within its AOI for a given node. Enclosing neighbors are nodes whose regions immediately surround the given node, and boundary neighbors are AOI neighbors whose enclosing neighbors may partially lie outside the AOI (Figure 2-3 (b)). Each node maintains a Voronoi diagram of all AOI neighbors and directly connects them to minimize latency.



*Voronoi diagram: a) dots indicate sites, and lines define boundaries (**edges**) for **regions**; b) the 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.*

Figure 2-3：Voronoi diagram [11]

As only a few neighbors are kept, the cost to maintain a Voronoi diagram at each node is low. It just requires each node to minimally keep its enclosing neighbors to prevent overlay partition. When a node moves, position updates are sent to all connected neighbors. According to knowing this moving node and other nodes beyond the AOI, neighbor discovery is done via notifications from boundary neighbors. Using this way, potential AOI neighbors are discovered with mutual collaborations. As a node moves around, it will constantly discover new nodes and disconnect those that have left its AOI. A node thus restricts communications with mostly the actual AOI neighbors, independent of the scale of the system. Keeping bandwidth consumption at each node bounded is the key to scalability of VON. To achieve true scalability, P2P-NVE must be able to constrain resource use at each node. VON is an elegant solution that organizes neighboring nodes with Voronoi diagrams, and realizes neighbor discovery through mutual node collaborations. As given above results, it is a scalable, efficient, and simple method in P2P-NVE establishment.

After the neighbor discovery, the next step of construction of P2P-NVE is the content delivery. The way to the progressive content transmission, or called 3D streaming [23] [24], is necessary. Similar to audio or video media streaming, 3D content needs to be fragmented into pieces at a server, before it can be transmitted, reconstructed, and displayed at the clients. Because users accessing 3D content often have different visibility or interests, transmission sequence is a key point. Unlike media streaming, 3D streaming thus varies from user to user and requires individualized visibility calculations [25]. Although current 3D streaming schemes can be classified into four main types: object streaming, scene streaming, visualization streaming, and image-based streaming [26]. The scene streaming is major issue to be dealt with in the Flowing Level-of-Details (FLoD) [27].

FLoD is a framework for P2P-NVE scene streaming, which usually involves a collection of 3D objects placed arbitrarily in space that are streamed to clients according to user visibility or interests. The main design rationale of FLoD is when users in large-scale P2P-NVE, a node might have overlapped visibility with its AOI neighbors. It is thus likely that the neighbors already possess relevant 3D content. The server can be relieved from serving the same data repetitively if nodes request data from the neighbors first. Note that neighbors here are proximity on the virtual map, not on the physical network.

Recent research on P2P-NVE overlay allows AOI neighbors' information such as IDs, virtual coordinates, and IP addresses can be learned without relying on a server. Based on VON, a kind of P2P overlay, it can be efficient in FLoD. As a node moves around, given a position and AOI-radius, it can constantly notify the overlay of its position and get refreshed information on AOI neighbors. The discovery of AOI neighbors is the discovery of the proper interest groups for distributed content sharing.

To economically distribute scene descriptions to clients, FLoD partition the P2P-NVE into fixed-size square cells. It is similar to Cyberwalk [28]. Each cell has a small scene description specifying the objects within. Each 3D object is specified by a unique ID, location point, orientation and scale within the scene description. Determining the visible objects to retrieve can thus be done in a fully distributed manner, as each node is able to locally determine the cells covered by its AOI.

In Figure 2-4, the red circle is the AOI of the star node, and triangles are other user nodes. Various shapes are the 3D objects, with their location points as dots. Note that cell IDs can be calculated given the star node's location coordinate, the world dimensions and cell size. When entering a new area, a client first prepares a scene request list to obtain scene descriptions from its AOI neighbors or the server. Once scene descriptions are obtained, the client then judges which objects are in view and produces a piece request list to request for visible objects. Piece dependency, if any, is also specified in the piece request list to ensure that data retrieval adheres to the correct piece ordering for object reconstructions. Views are rendered progressively as data pieces arrive from either the peers or the server which acts as the final data source if peers cannot fulfill the requests. This iterative process of requesting scene descriptions and object pieces is repeated continuously as a client moves in the P2P-NVE.

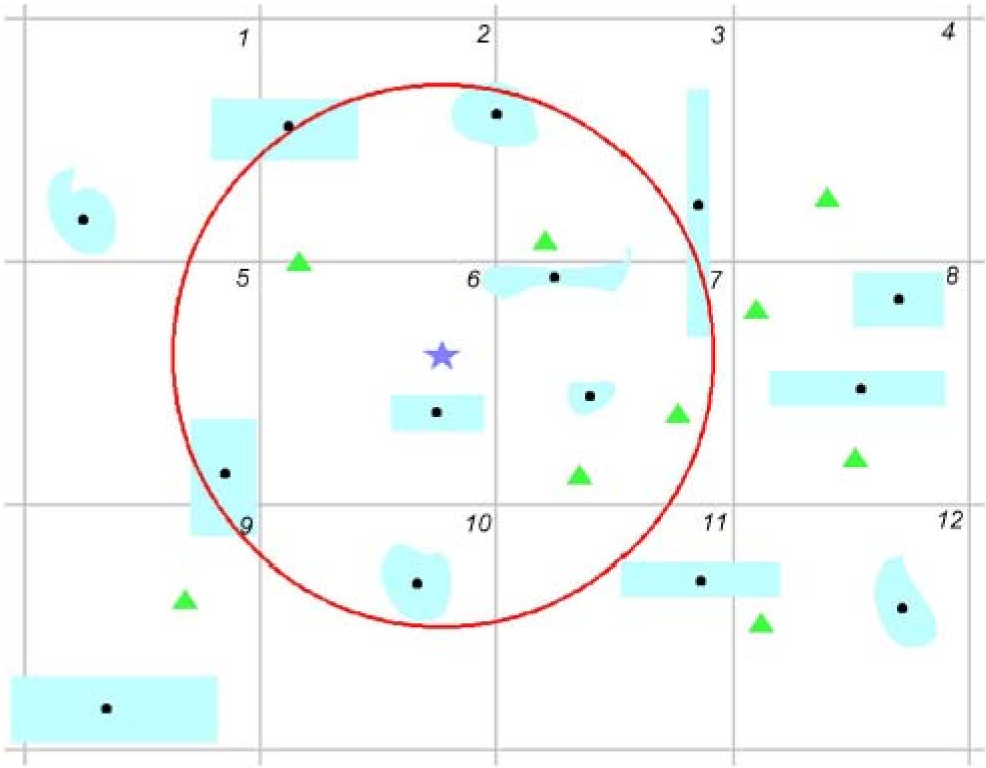
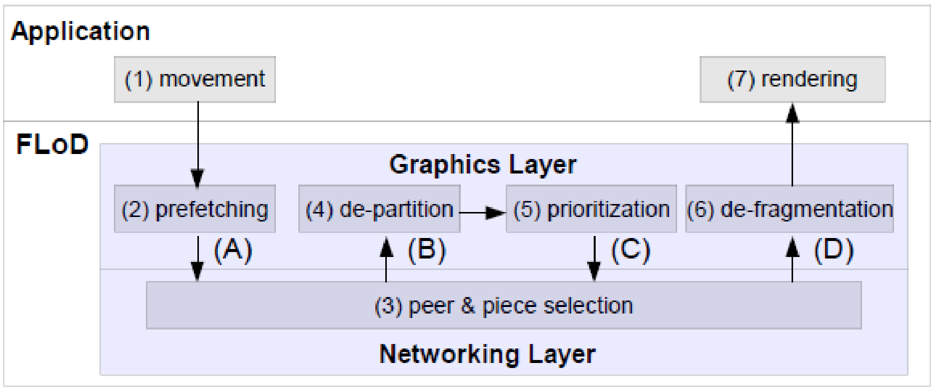


Figure 2-4：The overview of FLoD [26]

The following description is more detail procedure in FLoD. FLoD separates the main client-side tasks into a graphics layer and a networking layer. The graphics layer performs object determination (Figure 2-5 (2) (5)), and object reconstruction (Figure 2-5 (4) (6)), while the networking layer is responsible for object transmission (Figure 2-5 (3)). The application sits on top of FLoD and performs the usual tasks of taking user movement commands and performing rendering (Figure 2-5 (1) (7)).



FLoD's client-side task flow and layers. Data flows: (A) scene request list (B) scene descriptions (C) piece request list (D) data pieces. The numbers are task labels in FLoD's *Procedures*.

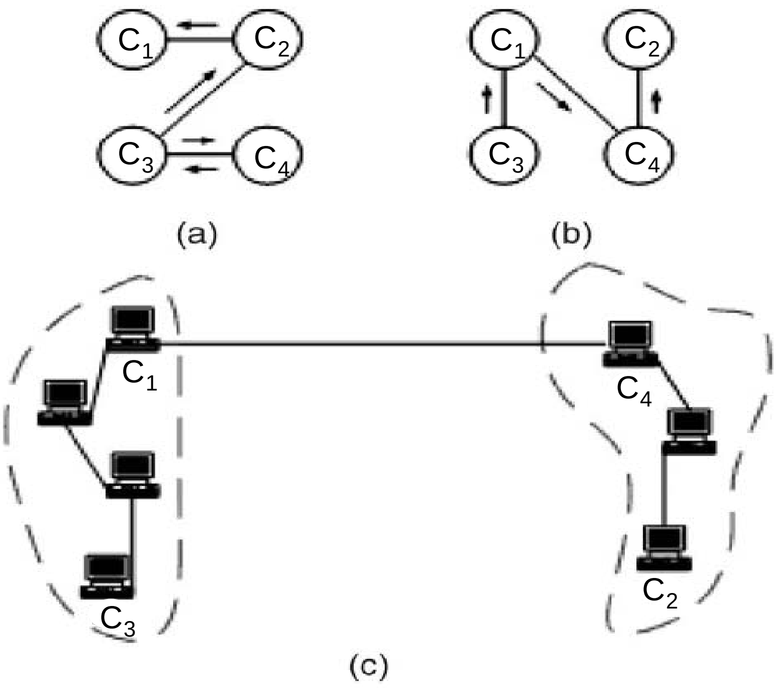
Figure 2-5：The detail procedure [27]

2.2 Locality-Aware

As the discussion of above section, AOI neighbors here are proximity on the virtual map, not on the physical network. The distance of AOI neighbors is near but may be far in the real world. Analogously, it causes an issue to the conventional problem called mismatch [29] in P2P architecture. The problem is that if peers randomly choosing logical neighbors without any knowledge about underlying physical topology can induce a serious topology mismatch between the P2P overlay network and the physical underlying network.

For example, Figure 2-6 (a) and Figure 2-6 (b) are two overlay topologies on top of the underlying physical topology shown in Figure 2-6 (c). Suppose C1 and C3 are in the same autonomous system (AS), while C2 and C4 are in another AS. We assume that the physical link delay between C1 and C4 is much longer than all of the other links in Figure 2-6 (c). Clearly, in the inefficient overlay of Figure 2-6 (a), the query message from source S will traverse the longest link C1C4 four times, which is a scenario of topology mismatch. If we can construct an efficient overlay shown in Figure 2-6 (b), the message needs to traverse all the physical links in Figure 2-6 (c) only once.

The inefficient topology brings great redundant latency in any Internet infrastructure. Therefore, there are some proposals intending to solve this problem. Location-aware topology matching (LTM) [29], adaptive connection establishment (ACE) [30], and scalable bipartite overlay (SBO) [31] are three of them. However, the peers don't connect to each other in P2P-NVE and interest in the contents inside AOI only. These proposals are not suit our goal and cannot deal with our problem totally.



Topology mismatch problem. (a) Inefficient overlay. (b) Efficient overlay. (c) Underlying physical topology.

Figure 2-6：Topology mismatch problem [29]

The problem we explore here is to find the nearby physical neighbor in P2P-NVE. Consistent hash [32] is easy to implement using existing network protocols such as TCP/IP, and particularly designed for use with very large networks such as the Internet. In video streaming, Liu [33] uses a two-level DNS to realize the directory service. The first level is the top-level domain (TLD), which could be any legal domain server. It contains location information for video servers and delegates its sub-domain to each video server. The second level is located at a video server, which contains location information of the proxy cache servers.

For example in Figure 2-7, suppose a user request Cj wants to access a video Ri. In Liu's method, the client first uses the video name as a key to compute a hash value H(Ri), and then connects to the TLD serve to get video server location Si. Once the client request has been received, the video server computes the table index of H(Cj), and returns Pj, which is the cache server closest to the user.

The method of consistent hash has the advantages of server load balance and system scalability. But it is forced to establish the hash table in advance and considered that the distance between node and node only by IP address. The lack of immediate measurement data between IP and IP can't actually reflect the existing network situation. Occasionally, the nearby IP is not the guarantee of the short distance in the real world.

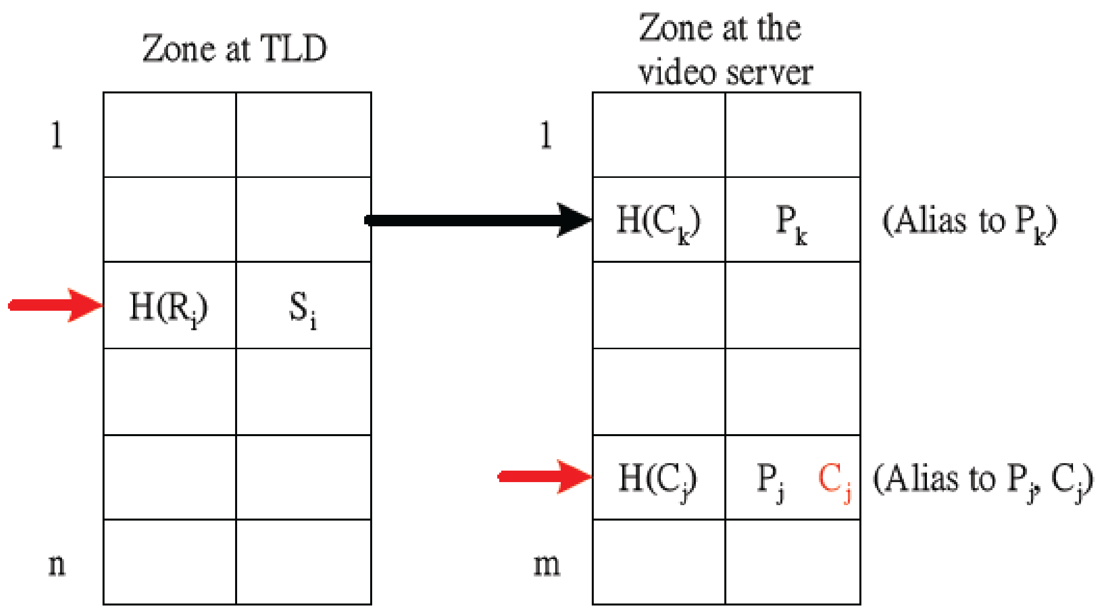


Figure 2-7：Consistent hash [33]

Our target is the short distance of physical neighbor discovery in P2P-NVE. The distance can refer to network latency in the Internet including P2P architecture. Two popular ways of measurement the latency in the network are hop counts and round-trip time (RTT). The first metric we considered is hot counts. In [34], allows a distance to be inferred between node and node given measurements from a measurement beacon to each. The rationale adopted by [34] is developed in [35]. Hotz [35] defines an N-tuple $< s_1, s_2, s_3, \ldots, s_N >$ as the distance to node from each of N measurement beacons. And then defines the distance a server at coordinates $S = < s_1, s_2, s_3, \ldots, s_N >$ and a client at $C = < c_1, c_2, c_3, \ldots, c_N >$ with the AVG function:

$$AVG(S , C) = (MAX(S , C) + MIN(S , C))/2$$

Where MIN and MAX are defined as:

$$MIN(S , C) = \max(|s_1 - c_1| , |s_2 - c_2| , |s_3 - c_3| \ldots |s_N - c_N|)$$

$$MAX(S , C) = \min(s_1 + c_1 , s_2 + c_2 , s_3 + c_3 \ldots s_N + c_N)$$

and min and max are the usual arithmetic minimum and maximum.

The intuition is illustrated in Figure 2-8. As measured from B, S could be located either at s' or at s", and C could be either at c' or c". The distance between S and C is therefore bound by |C – S| and C+ S, Generalizing to N beacons produces the MIN and MAX functions above. While these provide upper and lower distance bounds, Hotz showed that the AVG function generally produces a better result than either MIN or MAX. The value of AVG(S , C) represents the distance between S and C.

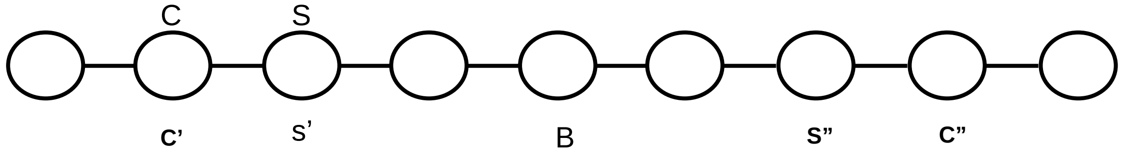


Figure 2-8：Motivation for Hotz' Distance Metric [34]

However, the geographic distance sometimes doesn't equal the network distance. The main purpose we wanted is how fast the service or content can be delivered. The other distance metric, RTT, takes account of both the network delay and server processing delay. In practice, it is more efficient using the mechanism of RTT to estimate the latency between node and node [36]. Several approaches are proposed to calculate the network distance by RTT as follow.

Global Network Positioning (GNP) [37] models the Internet as a geometric space and distributedly computes geometric coordinates to characterize the positions of the hosts in the Internet. It propose a two-part architecture in which a small distributed set of hosts called landmarks first compute their own coordinates in a chosen geometric space. These coordinates are then disseminated to any host who wants to compute its own coordinates relative to the coordinates of the Landmarks. Figure 2-9 (a) illustrates these Landmark operations for 3 Landmarks in the 2-dimensional Euclidean space. Figure 2-9 (b) illustrates these operations for an ordinary host in the 2-dimensional Euclidean space with 3 Landmarks. However, the computation of geometric coordinates is complex. This imperfection is not on demand for us in the field of network distance calculation.

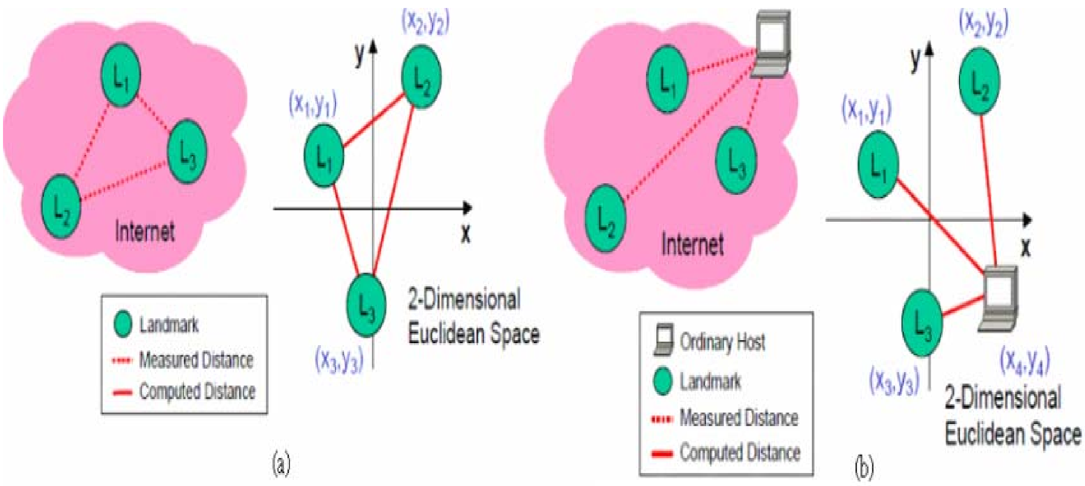


Figure 2-9：(a)Landmark operations (b) Ordinary host operations [37]

There is a rough but simple distance measurement in a manner. Shieh's method [38] is the worthful accomplishment in finding nearest neighbors in replication-aware CDN-P2P architecture by binning. Binning scheme [39] is an approach whereby nodes partition themselves into bins such that nodes that fall within a given bin are relatively close to one another in terms of network latency. This distributed binning scheme incorporates into the design of distributed systems such as overlay networks and content distribution systems. Binning scheme requires a set of well known landmark machines spread across the Internet. An application node measures its distance by RTT to this set of well known landmarks and independently selects a particular bin based on these measurements. A node's RTT measurements to each landmark offer two kinds of information: the first is the relative distance of the different landmarks from the given node and the second is the absolute value of these distances. Illustrated in Figure 2-10:

The relative distance: augment the landmark ordering of a node with a level vector; one level number corresponding to each landmark in the ordering. Its distance to landmarks L_1, L_2, and L_3 are 232ms, 51ms and 117ms respectively. Hence its ordering of landmarks is " $L_2L_3L_1$".

The absolute distance: divides the range of possible latency values into 3 levels; level 0 for latencies in the range [0,100]ms, level 1 for latencies between [100,200]ms and level 2 for latencies greater than 200ms. Using the 3 levels defined above, node A's level vector corresponding to its ordering of landmarks is "0 1 2".

Thus, combining with the relative and absolute distance, the result of node A's bin is "$L_2L_3L_1$: 0 1 2 ".

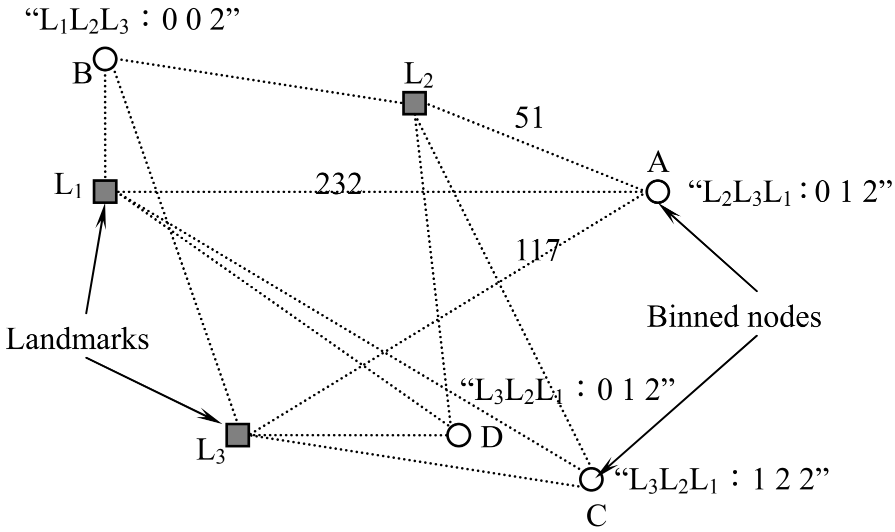


Figure 2-10：Distributed binning [39]

In conclusion, the above binning scheme does a reasonable job of placing nearby nodes into the same bin, such as node C and node D. In other words, node C and node D represent a set of approximate physical neighbor. Binning is simple, requiring very little from measurement and infrastructure. Also is very scalable because nodes independently discover their bins without communicating or coordinating with other application nodes. It is a practical mechanism to gather location information effectively. Given the above remarkable binning strategy, we apply this binning strategy to our method of approximate physical neighbor discovery in P2P-NVE.

Chapter 3

METHOD

3.1 System Model

Let’s imagine a circumstance when a user is entering and exploring in a NVE such as MMOG. Everything is vague and limitary including scenes, objects, anime, people, environment map, etc. This user must download content before the application program can render the frame. However, as the content becomes larger and larger, it takes more time to wait before the downloading is complete. In the case of MMOG, most of the 3D content is widely distributed by the pre-installation CDs nowadays. Nevertheless, players still have to download lots of data while new update is release. The more players connect to the server and download the 3D content, the more waits players may suffer because of the heavy loading of server. In order to eliminate this uncomfortable experience for players, how to save the total downloading time and allow the user to see the scene in NVE as early as possible is important.

The basic strategy is to compress the 3D data and transmit them progressively [40]. As long as the size of 3D data is decreased, and the time and bandwidth will be saved when content delivery. If the 3D data can be transmitted progressive, the user can have a glance of the virtual world as soon as possible. Although the visual quality is not very good in the first place, it will become better and better after more 3D data is received from providers.

Besides the issue of content size, while the more players join in the NVE, the lack of scalability is another major problem because of the client-server architecture. One new class of network architecture called P2P is emerged and adopted to take up the positions of the fallen and rise to fight one after another. It is excellent that total resource can be increased with each entering of the new node. This feature makes more and more users participating in the NVE without server resource expanded. Therefore, it is desirable to integrate P2P with NVE.

In this modern architecture, P2P-NVE, a server is still needed to contain the all information of the entire network virtual environment, but most of its work loads can be bypassed to each user who participating in P2P-NVE. Thus, how to discover nodes called neighbors appropriately and to deliver content as soon as possible are important. FLoD [27] is the success framework for P2P-NVE, in here each user can explore the neighbors through AOI and then almost downloads desirous of 3D content without server. To take the notice of it, neighbors we called here are close to each other in the NVE only. While peers just choosing neighbors based on logical relationship but without any knowledge about physical network information will be the new coming problem. The content will be transferred probably through the far distance or long latency in the real world. It may induce the serious waste and delay in the P2P-NVE.

Today's Interest can be viewed as a collection of interconnected routing domains, which are groups of nodes that are under a common administration and share routing information [41]. The previous researches [38] [39] also point out the latency among nodes and nodes can be very different and very lengthy in worse case. Illustrated in Figure 3-1, Transit-Stub (TS) topology models network using a 2-level hierarchy of routing domains with transit domains that interconnect lower stub domains. In this TS topology, the link latencies are 100ms for inter-transit domains, 20ms for intra-transit domains, 5ms for stub-transit domains, and 2ms for intra-stub domains.

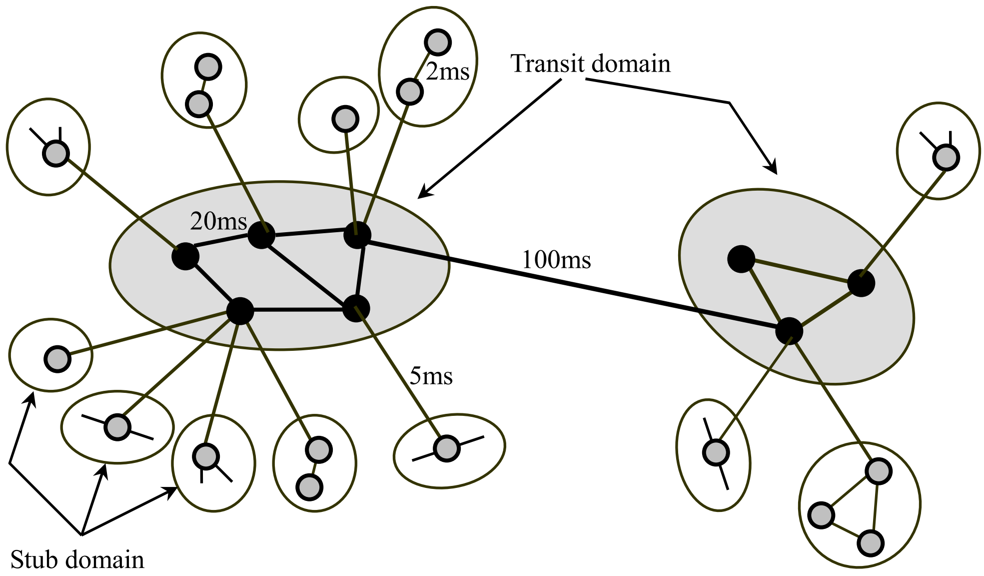


Figure 3-1：Internet domain structure [38]

We assume that it is efficient when the data cannot only be received from the close virtual neighbors, but also be got accurately by choosing the nearby physical neighbors. Furthermore, it is possible that user is just walking around and wants to have a glance of the virtual world. The first priority is delivering the content to user as soon as possible. This is the notable one of the characteristic of the P2P-NVE. To find the nearest physical neighbor sometimes is unnecessary. The complicated computing and full 3D content downloading are cost effective. Preciseness and rapidity is a kind of trade-off. Therefore, we propose the metric of the approximate physical neighbor discovery in the P2P-NVE.

After the above observation and discussion, we will list the component and its functions in our system design at the following description. Brief explanation of our concept is that if each part takes his responsibility and then cooperates to the mate smoothly, the high availability of the system is accomplished.

The architecture in our method is composed of the content servers and the peers:

1. Content server(s)
    I. The provider with all content
    II. Accept the delivery request from all peer
    III. Send the content to all peer
    IV. The role of landmark
    V. Allocate the new peer to the group
    VI. Receive, pass, and handle the message from peer
    VII. Adjust the tree architecture inside the own group if necessary
2. Peer(s)
    I. Send the content to other peer
    II. Accept the delivery request from other peer
    III. Retain and refresh the AOI neighbor
    IV. Hold the grandfather, father and child peer table
    V. Maintain the tree architecture inside the belonging group
    VI. Send the query message to peer
    VII. Respond the query message from peer

In Figure 3-2, it is the overview of our system model. The up part is the brief diagram of FLoD simulation application. It describes the segment of the scenes, the position of the objects, and the situation of the peers in the virtual environment. The under part is the result after our method operated. All peers will be separated into sever group with each short latency. Finally, the arrow is the mapping of the same peer from the virtual environment to the real world.

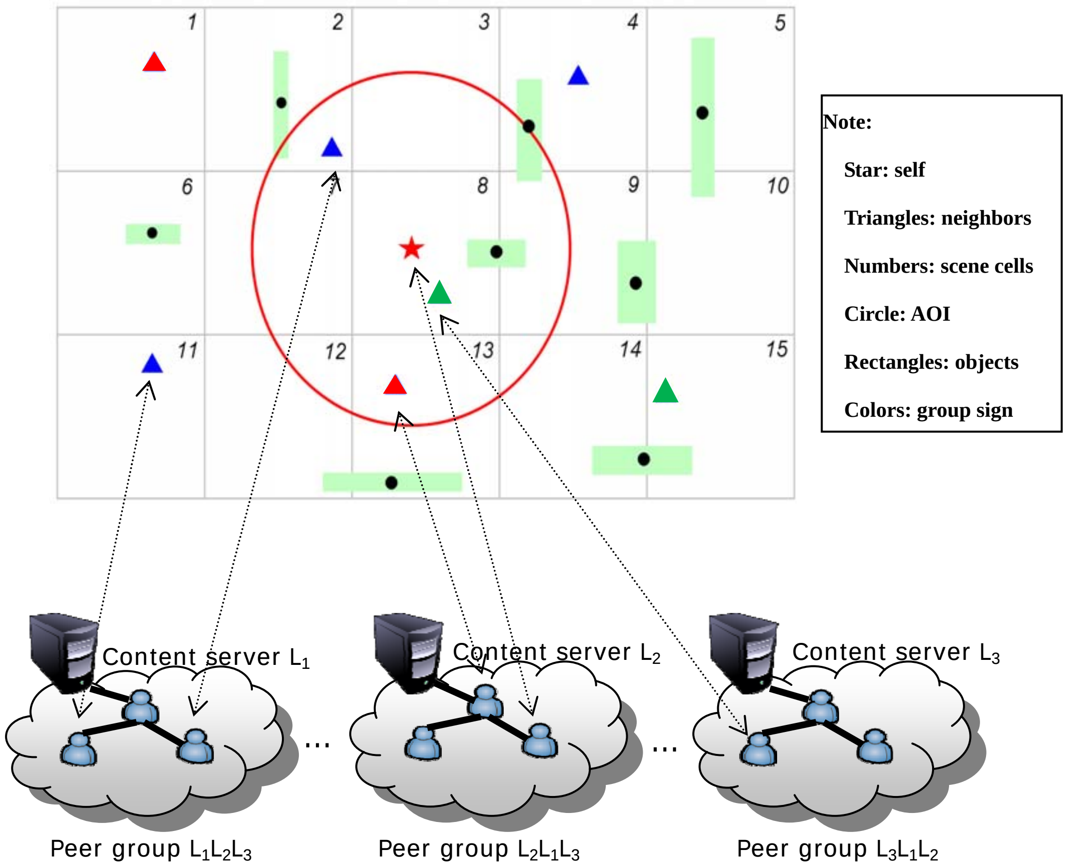


Figure 3-2：System overview

3.2 Our method

In Figure 3-3, it is the flow of our method here called Discovery of Physical Neighbor (DPN). First step expresses the way of grouping by the server — the creation of grouping. Second step is how to arrange and maintain peer with tree architecture — the preservation of grouping. The following sequences are detail descriptions about first step and second step.

(a). End user chooses any NVE to enter

(b). End user logins to the content server

(c). End user be pinged with the content servers (landmarks) respectively and gets the peer group number by binning.

(d). The content server distributes the end user to his peer group.

(a) ~ (d) is the first step of the flow of DPN.

(e). End user joins to his peer group by tree architecture.

(f). Every peer pings his father peer periodical in his group.

(g). If the leave of peer is happened, peers maintain the tree architecture.

(h). If no peer leave, keep the tree architecture static.

(i). Delivers content to each peer.

(e) ~ (i) is the second step of the flow of DPN.

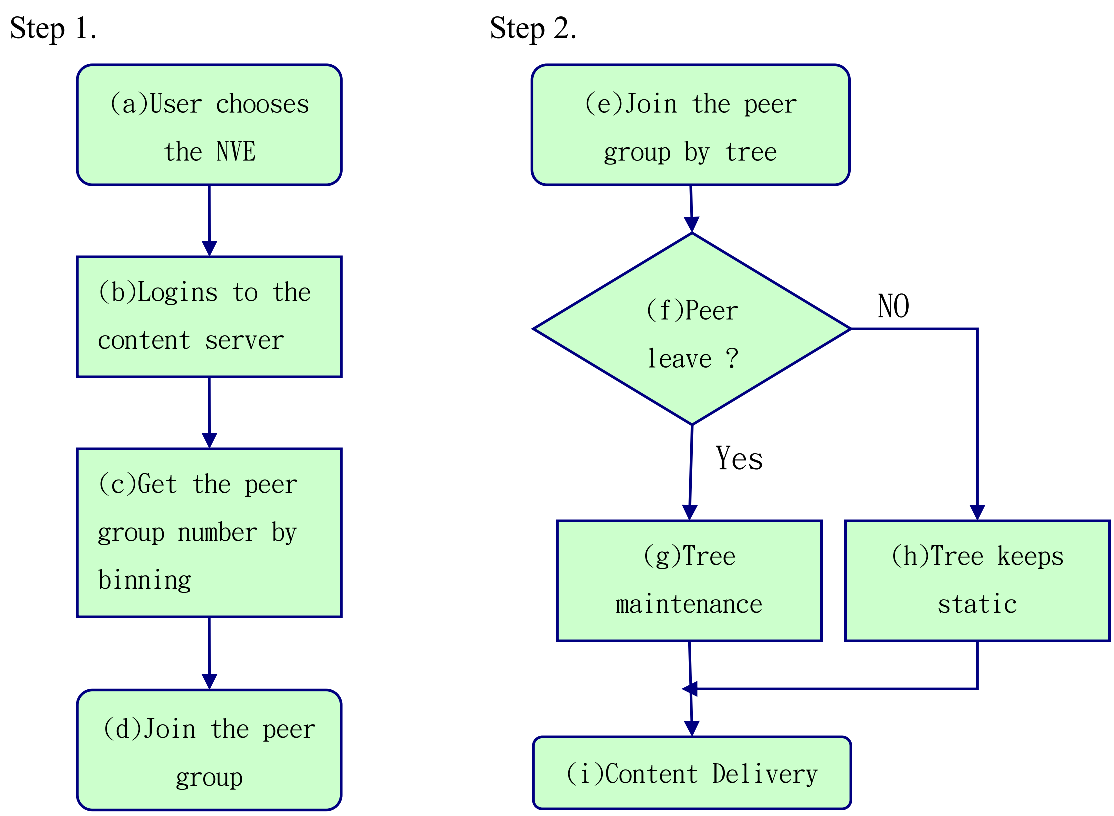


Figure 3-3：The flow of DPN

3.2.1 The creation of grouping

In this section, we introduce the process of the creation of grouping. First, the location of landmarks must be considered. How many landmarks exist and how distributed they located are two important issues. There are several ways to pick the suitable landmarks addressed in these previous methods [37] [38] [39]. Suppose that there are m servers distributed everyplace, and then pick n servers from m servers to perform in the role of landmarks ($m \geq n$).

(a). Maximum Separation: the largest sum of distance among all servers.

(b). M-Medians: the shortest sum of distance among the servers picked to become landmarks and the servers not picked to become landmarks.

(c). N-Cluster-Medians: it makes the m close servers to several clusters, and then picks each central server inside each cluster to become landmark.

(d). Random: it randomly picks the n servers from m servers.

In order to simplify and accelerate the system, we set the content servers to be the landmarks directly and assume these landmarks are well-distributed. And then all peers will be divided to n! groups at most if there are n content servers to become the landmarks. Each user will have the results of Round-Trip Time (RTT) with all landmarks by pinging. After collecting and sorting the results of RTT, content server will assign a group to end user by binning.

As showed in Figure 3-4, there are three content servers L_1, L_2, and L_3 to be landmarks. After sorting the results of RTT, there will be separated to A：$L_2L_1L_3$, B：$L_1L_2L_3$, C：$L_1L_3L_2$, D：$L_2L_3L_1$, E：$L_3L_2L_1$, and F：$L_3L_1L_2$ six group at most. The results of RTT of new peer between he and all three landmarks are RTT(R_1 , L_1)：70ms, RTT(R_1 , L_2)：210ms, and RTT(R_1 , L_3)：127ms. Then the content server sorts the results and gets relative distance $L_1L_3L_2$ of this new peer. Hence, the new peer will be assigned to C：$L_1L_3L_2$ group. According to this scheme of binning, we can partition all peers to several groups automatically. Eventually, peer 1 is assigned to F：$L_3L_1L_2$ group, peer 2 is assigned to E：$L_3L_2L_1$ and peer3 is assigned to A：$L_2L_1L_3$ respective.

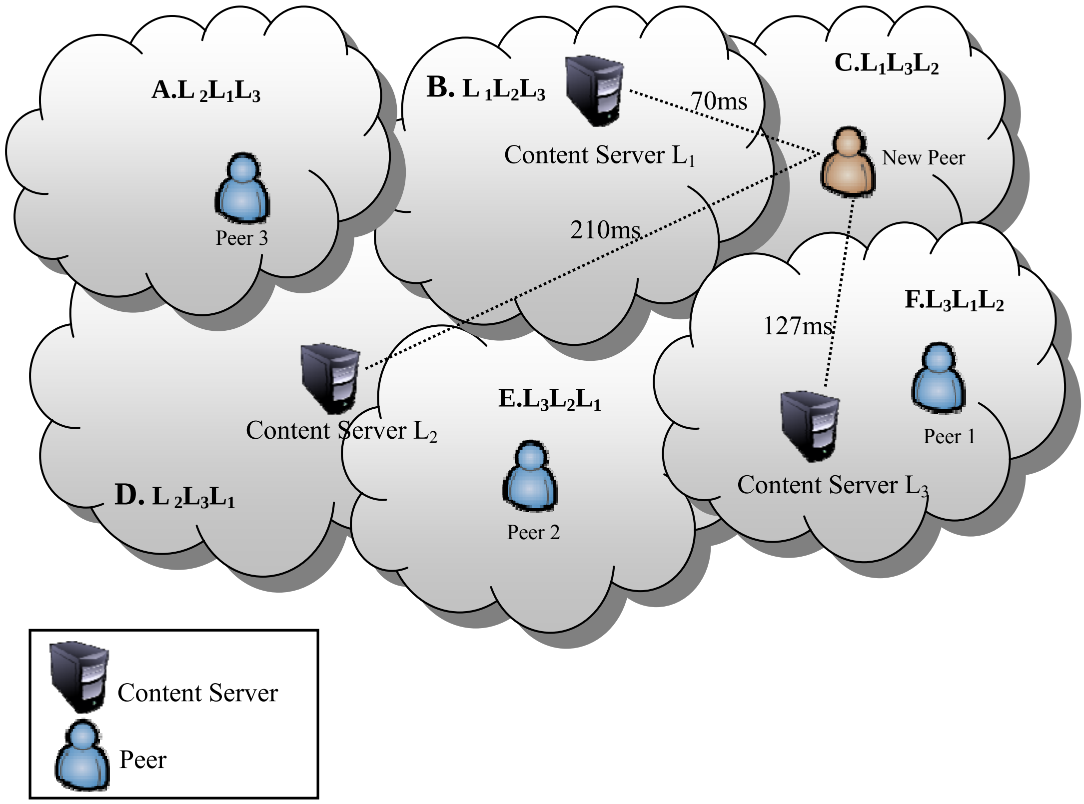


Figure 3-4：The configuration of grouping

Figure 3-5 is the process of the binning step by step. When new peer joining the P2P-NVE (a), content server $landmark_1$ which new peer connected will ask the other content server to ping this new peer (b). And then each content server echoes the request to the new peer and does the RTT measuring with the new peer (c). After a short period, the content server will receive the results of RTT of new peer (d). As a result, the content server $landmark_1$ can assign the new peer to group by binning (e).

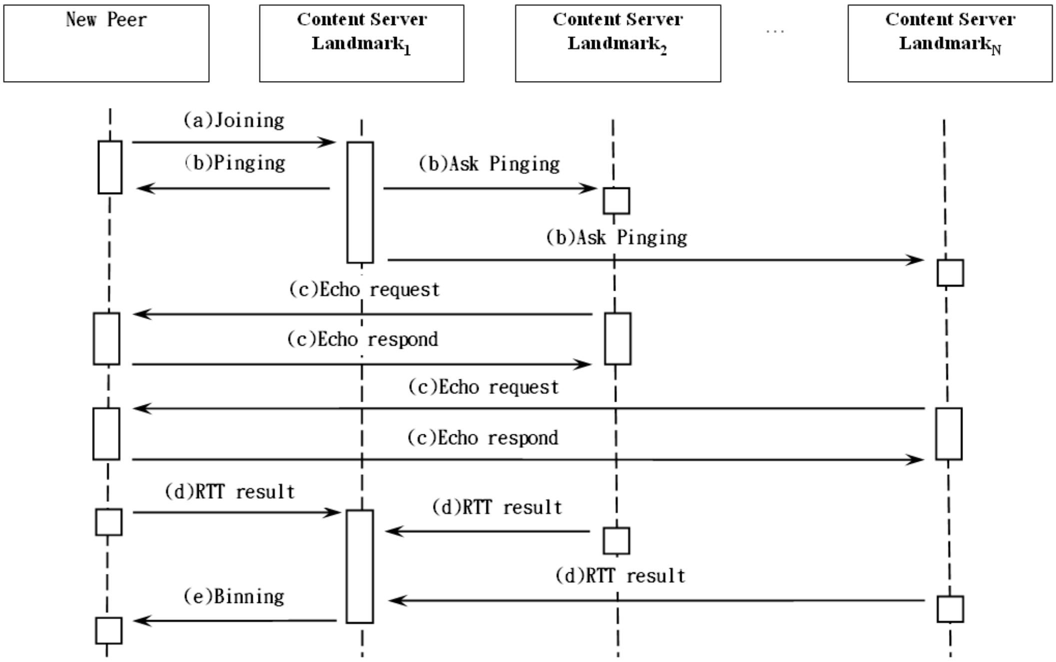


Figure 3-5：The process of binning

According to the analysis of player distribution in [42], we can realize that the number of concurrent players (M) in the MMOG fits the Weibull distribution (1). In this paper, right after a game world is launched, few players would be aware of it until some time later. Therefore, the number of concurrent players in these time slots is a small value. After removing these starting points, it is found that the distribution can be depicted perfectly in this distribution, where α = 5.1828, β = 0.081.

$$P(x) = \alpha\beta^{-\alpha}x^{\alpha-1}e^{-(x/\beta)^{\alpha}} \tag{1}$$

In the previous speaking, the n landmarks will divide players into n! groups. Assume each player is allocated into each group with uniform distribution, and then the number of concurrent players (Q) in a group can be denoted:

$$Q = M/n! \tag{2}$$

3.2.2 The preservation of grouping

While each group is successful creation, there are several procedures needed to be executed inside each group. Next, we will describe these procedures including joining, leaving and maintenance of group in the DPN. In the first place, peer performs the joining procedure after binning by server. Illustrated in Figure 3-6, the following descriptions are joining procedure:

(a). According to the relative distance of new peer, the content server sends the new peer to his peer group (step 1-1).

(b). The content server gives a exist node to new peer for connection using AVL-tree (step 1-2).

(c). The peer sends the connection request to this exist node (step 2-1).

(d). After receiving the reply, the peer is connecting to this node (step 2-2).

(e). The new peer creates the grandfather, father, and child nodes table (step 3-1).

(f). The father node and new peer exchange the information to each other for updating the table (step 3-2).

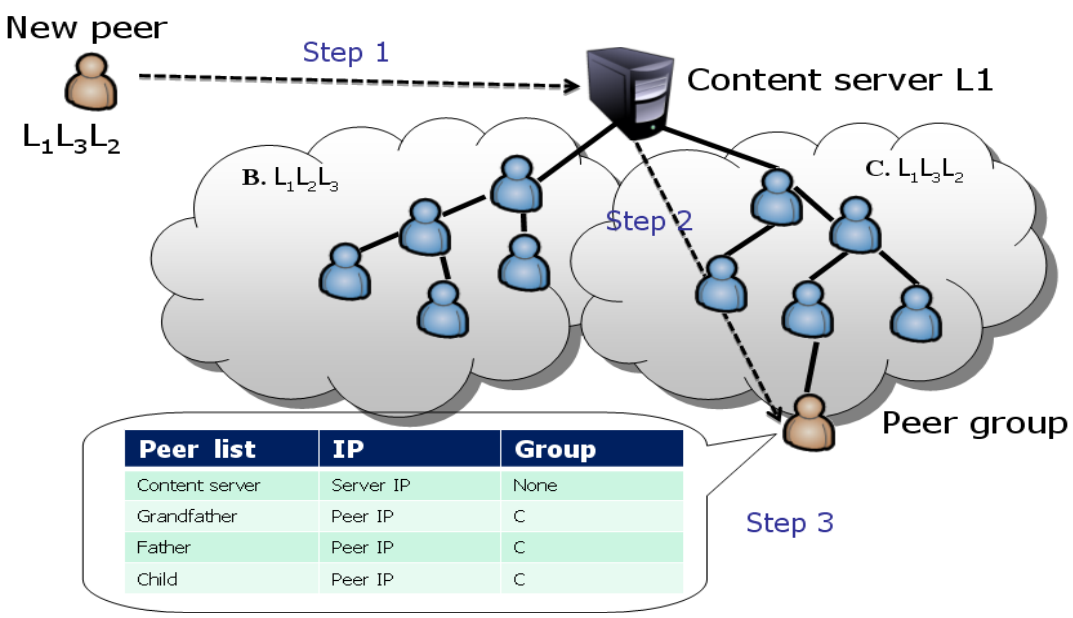


Figure 3-6：The joining procedure

The second description is the basic procedures for leaving. When proper departure event occurring, such as player logout, a leaving peer simply disconnect from the peer group. Illustrated in Figure 3-7, the following descriptions are leaving procedure:

(a). The leaving peer notifies the peers inside the table (step 1).

(b). The notified nodes modify the table (step 2).

(c). The leaving peer normally leaves (step 3).

(d). Content server adjusts the tree architecture using AVL-tree.

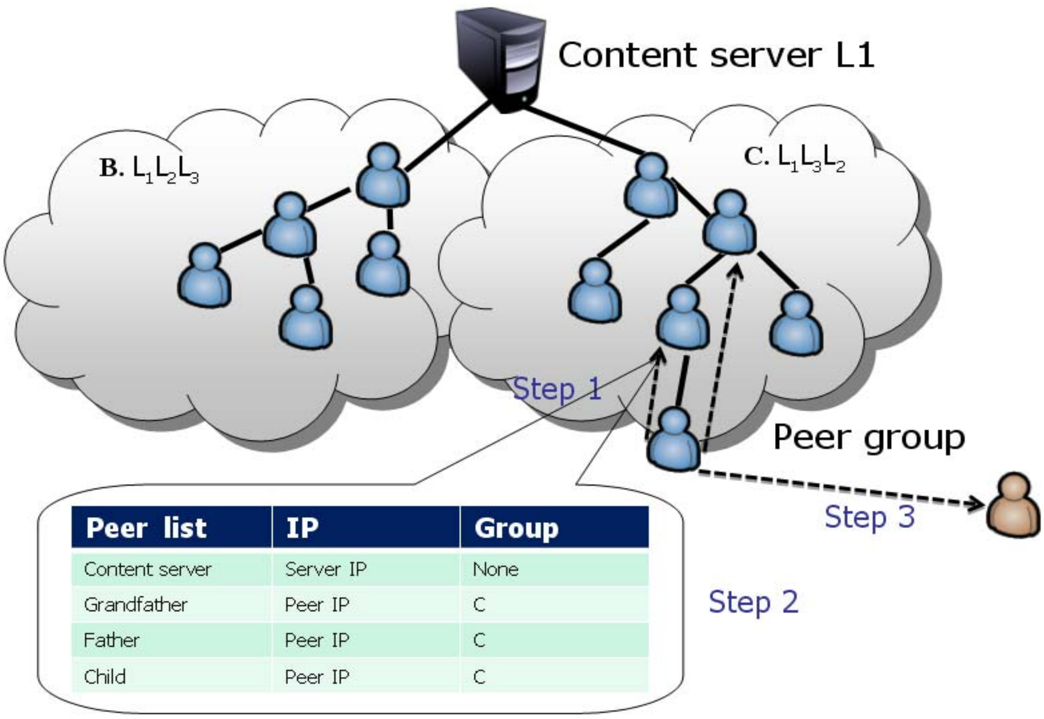


Figure 3-7：The leaving procedure

Finally description is the basic procedure for maintenance. In each group, every peer pings his direct father node in a periodic time (t). When the node is healthy, the tree architecture can remain completeness. If the direct father node doesn't reply, we consider this father node is invalid. Therefore, the maintenance procedure is appeared to handle this situation. The three kinds of condition for recovery are illustrated in Figure 3-8 (a), Figure 3-8 (b), and Figure 3-8 (c).

A. Direct father node is invalid:

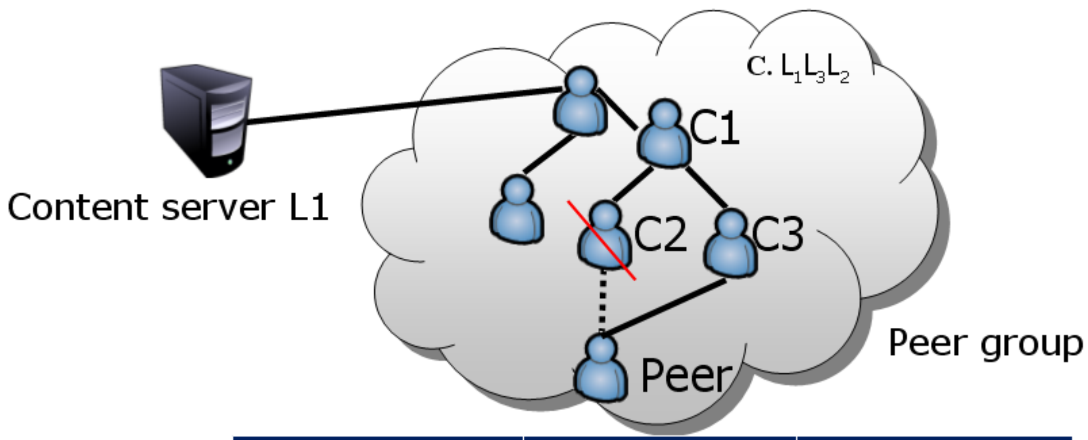
(a). The peer pings his direct father node periodic, and the direct father node is no respond after a waiting time.

(b). The peer looks for the other father node in his table and sends the connection request to the other father node.

(c). After receiving the reply, the peer is connecting to this father node.

(d). The peer, the content server, and his new direct father node exchange information to each other for updating the table.

(e). Content server adjusts the tree architecture using AVL-tree.



Peer list	IP	Group
Content server	Server	None
Grandfather	Peer C1	C
Father 1	Peer C2	C
Father 2	Peer C3	C

Figure 3-8(a)：The maintenance procedure A

B. All father nodes are invalid:

(a). The peer pings his direct father node periodic, and the direct father node is no respond after a waiting time.

(b). The peer looks for the other father node in his table and sends the connection request to the other father node.

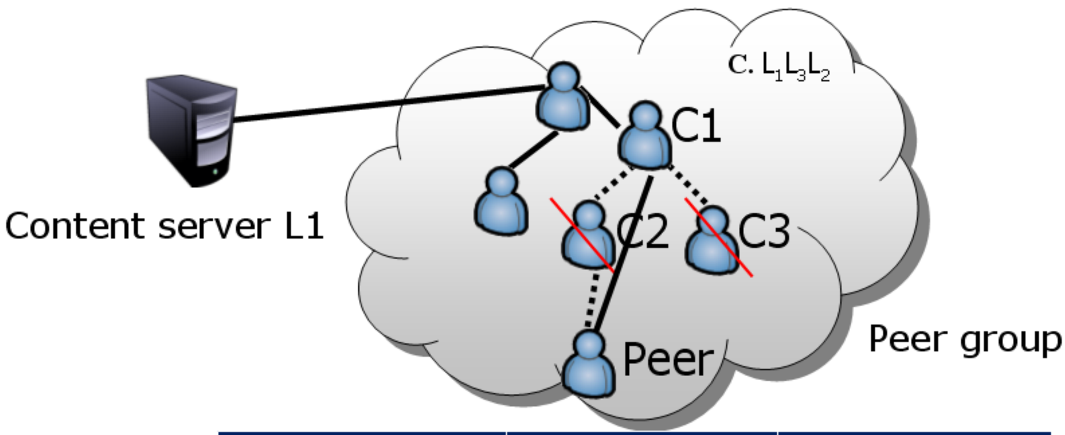
(c). The all other father node are also no respond after a waiting time.

(d). The peer looks for the grandfather node in his table and sends the connection request to the grandfather node.

(e). After receiving the reply, the peer is connecting to this grandfather node.

(f). The peer, the content server and his new direct father node exchange information to each other for updating the table.

(g). Content server adjusts the tree architecture using AVL-tree.



Peer list	IP	Group
Content server	Server	None
Grandfather	Peer C1	C
Father 1	Peer C2	C
Father 2	Peer C3	C

Figure 3-8(b)：The maintenance procedure B

C. All father nodes and grandfather are invalid:

(a). The peer pings his direct father node periodic, and the direct father node is no respond after a waiting time.

(b). The peer looks for the other father node in his table and sends the connection request to the other father node.

(c). The all other father node are also no respond after a waiting time.

(d). The peer looks for the grandfather node in his table and sends the connection request to the grandfather node.

(e). The grandfather node is also no respond after a waiting time.

(f). The peer sends the connection query to the content server he belonged to (step 1).

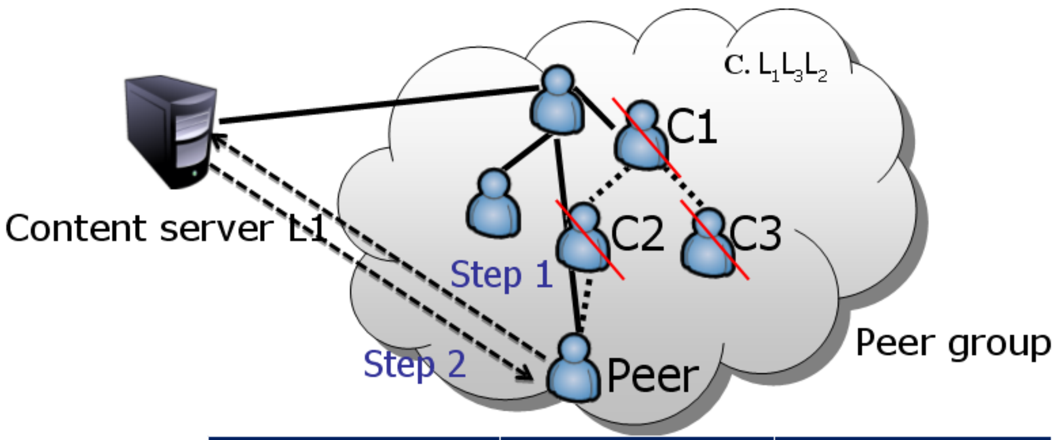
(g). The content server gives a exist node to the peer for connection using AVL-tree (step 2).

(h). The peer sends the connection request to this exist node.

(i). After receiving the reply, the peer is connecting to this node.

(j). The peer, the content server, and his new direct father node exchange information to update the table of each other.

(k). Content server adjusts the tree architecture using AVL-tree.



Peer list	IP	Group
Content server	Server	None
Grandfather	Peer C1	C
Father 1	Peer C2	C
Father 2	Peer C3	C

Figure 3-8(c)：The maintenance procedure C

Here we will discuss with the average time of recovery while the maintenance procedure executed. We define the symbol first in Table 3-1. Inside peer group, the peer pings his direct father node in a periodic time (t). In the first place, if the direct father node doesn't reply, we consider this father node is invalid and the average time of recovery about maintenance procedure A for the peer is:

$$R_{avg} = (t + 2D_n) \cdot (1 - P) \tag{3}$$

Second, if all father nodes don't reply, we consider all father nodes are invalid and the average time of recovery about maintenance procedure B for the peer is:

$$R_{avg} = (2t + 2D_n) \cdot P \cdot (1 - P) \tag{4}$$

Third, if grandfather node and all father nodes don't reply, we consider these nodes are invalid and the average time of recovery about maintenance procedure C is:

$$R_{avg} = (3t + 2D_s) \cdot P^2 \tag{5}$$

Finally, all situations are randomly appearance and the average time of recovery about total maintenance procedure can be combined with above three equations to the one:

$$R_{avg} = (t + 2D_n)(1 - P) + (2t + 2D_n) P (1 - P) + (3t + 2D_s) P^2 \tag{6}$$

Table 3-1：The symbol definition

Symbol	Definition
t	Time of periodic pinging
D_n	Average latency between nodes and nodes
D_s	Average latency between content server and nodes
P	Probability of invalid node
R_{avg}	Average time of recovery

3.3 Content delivery

After the system building up, the peer can move in the P2P-NVE autonomously. Each peer also has the responsibility of obtaining of the scene descriptions and object data. When walking around in the virtual environment, the peer can discover the virtual neighbors by AOI. Once the content delivery is needed, the neighbor selection is both considered with AOI in the virtual relation and DPN in the physical relation. It is probable a best choice that the concept of nearby neighbor is not only logicality but also physicality.

3.3.1 Algorithm

The Figure 3-9 is the draft of algorithm and the pseudo code of our algorithm for content delivery flow is below it. We define an N-tuple < G1, G2, G3, …, Gn > as the each peer group in the real world. And then defines the scene $S = < S_1, S_2, S_3, S_4, S_5, S_6 >$, the object $O = <O_{11}, O_{12}, \ldots, O_{21}>$, and the node $U = <U_1, U_2, U_{new}>$ in the virtual map. Finally, the meaning of symbol T_k is current time and T_{k-1} is past time. The concept of current or pass time is similar to the view of inside or not inside AOI.

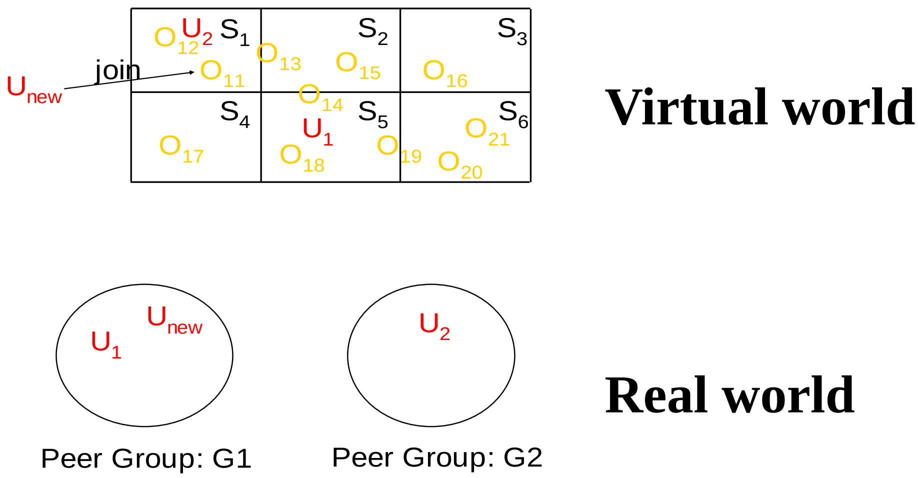


Figure 3-9：The draft of algorithm

Pseudo code:

```
U_new, U_1: {G1,G2,……Gm, Gn}=G1 ;     //U_new and U_1 are in the same group G1
U_2: {G1,G2,……Gm, Gn}=G2 ;            //U_2 is in the other group G2
while U_new join scene S, such as S_1
  if U_1 in scene S and own the object O, such as O_11,O_12,O_13 , in current time T_k then
      execute data transference from U_1 to U_new
  else if U_1 owned the object O in past time T_k-1 then
      execute data transference from U_1 to U_new
  else if U_2 in scene S and own the object O in current time T_k then
      execute data transference from U_2 to U_new
  else
      execute data transference from content server to U_new
  end if
end
```

3.3.2 Data streaming

In order to receive the content as soon as possible, we will classify the streaming priority in Table 3-2 first. Based on latency issue, the physical neighbor is more important than the virtual neighbor. Therefore, the peer in the same group means the shorter latency in the real world and must be chose with the high priority in DPN.

Table 3-2：The streaming priority

Priority	Group	AOI
1	The same	Inside
2	The same	Not inside
3	Different	Inside
4	Server	Server
5	Different	Not inside

Here is the diagram of the data streaming drew by the brief of FLoD simulation application. In Figure 3-10, assume the neighbors exist and own the content the peer needed. The peer will select the data streaming priority by the sequence number from 1 to 4. It notices that server is not showed in the diagram; therefore we don't consider it priority and give the sequence number to it.

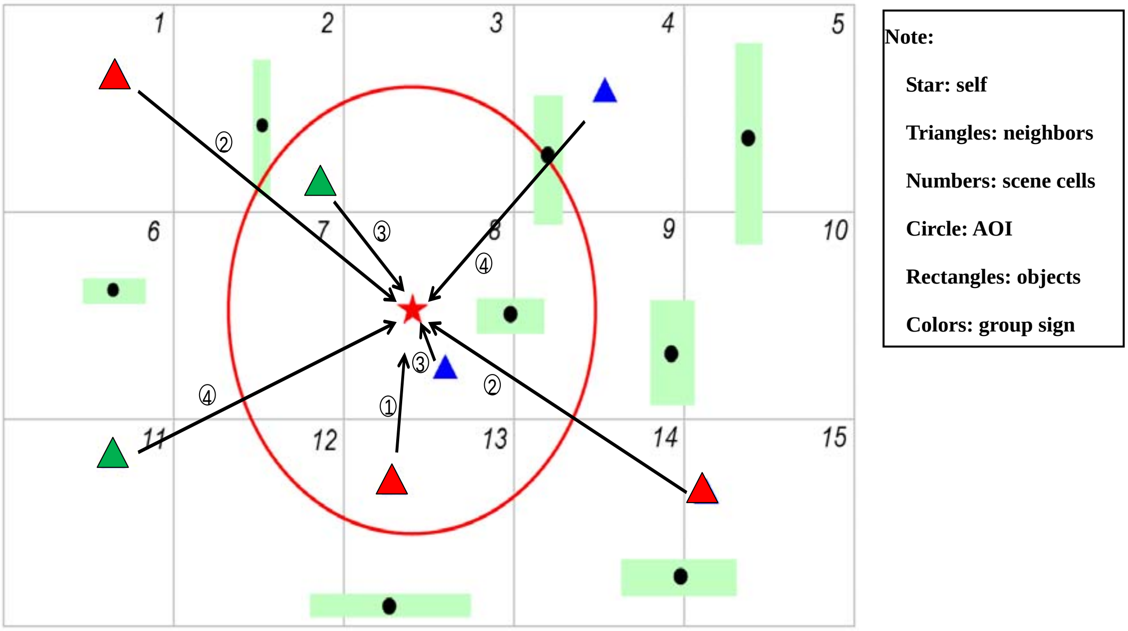


Figure 3-10：The draft of data streaming

The following is the other diagram of the data streaming considered from viewpoint of the objects. From this viewpoint, we can image that each object has its peer groups if a peer inside group own this object in the past. In other words, if a peer needs certain object content, it is like a peer joining to peer group owned by this object. In Figure 3-11, if the neighbors exist and own the content the peer needed, the peer will select the data streaming priority by the sequence number from 1 to 5. Because the position of server is dotted, the priority of server can be considered in the content delivery flow.

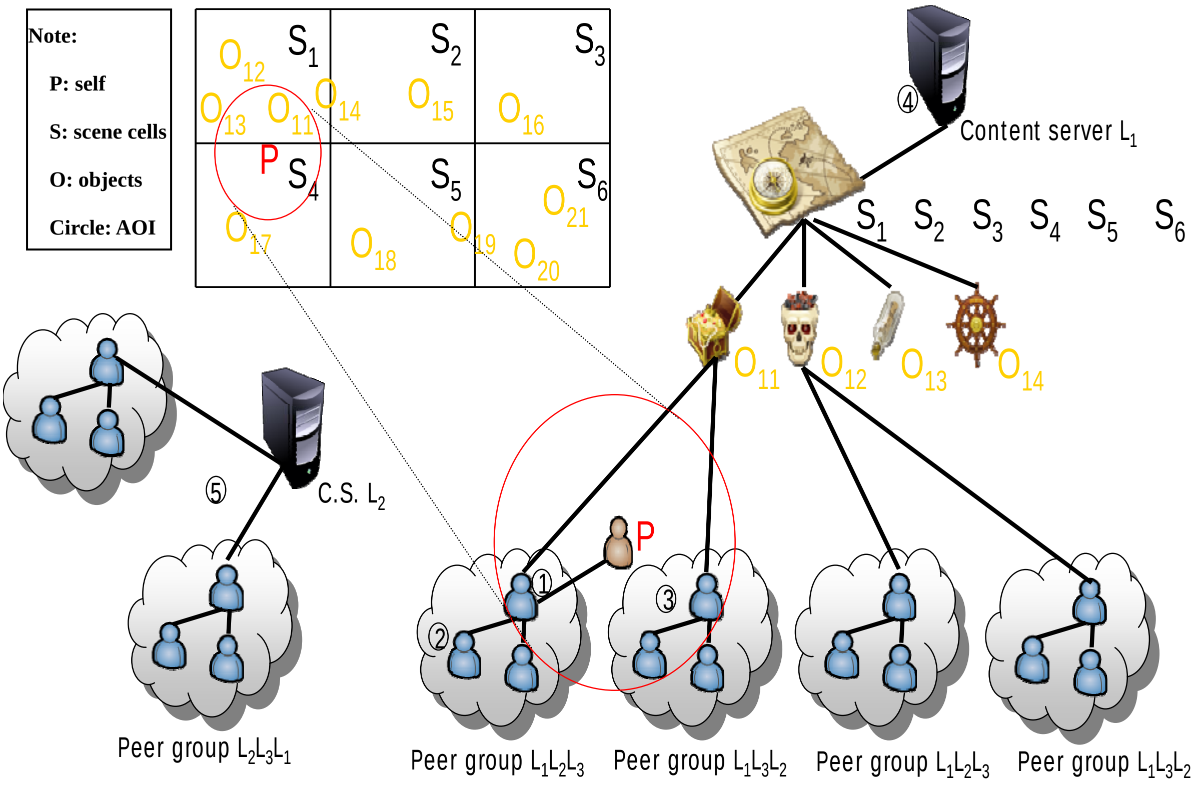


Figure 3-11：The diagram of data streaming

In Table 3-3, it is a table of the comparison between DPN and original FLoD. First column is the data source including the opinion of group and of AOI. It shows the capability of original FLoD can be improved by DPN further.

Table 3-3：The comparison

Data Source	DPN	FLoD
The same group	O	X
The different group	O	X
Inside AOI	O	O
Not inside AOI	O	X
Server	O	O

Because the peer group information is made by DPN, we can contact the other peers and receive the content from the different peer group via query message passed by the content server. However, according the efficiency issue, this behavior cannot be happened almost and has the lowest streaming priority as shown in the Table 3-2. It is much like the backup solution in the situation that if no peers have the desired content in the same group or inside AOI, and capacity or bandwidth of all content servers is in the peak. Finally, we illustrated it in the Figure 3-12.

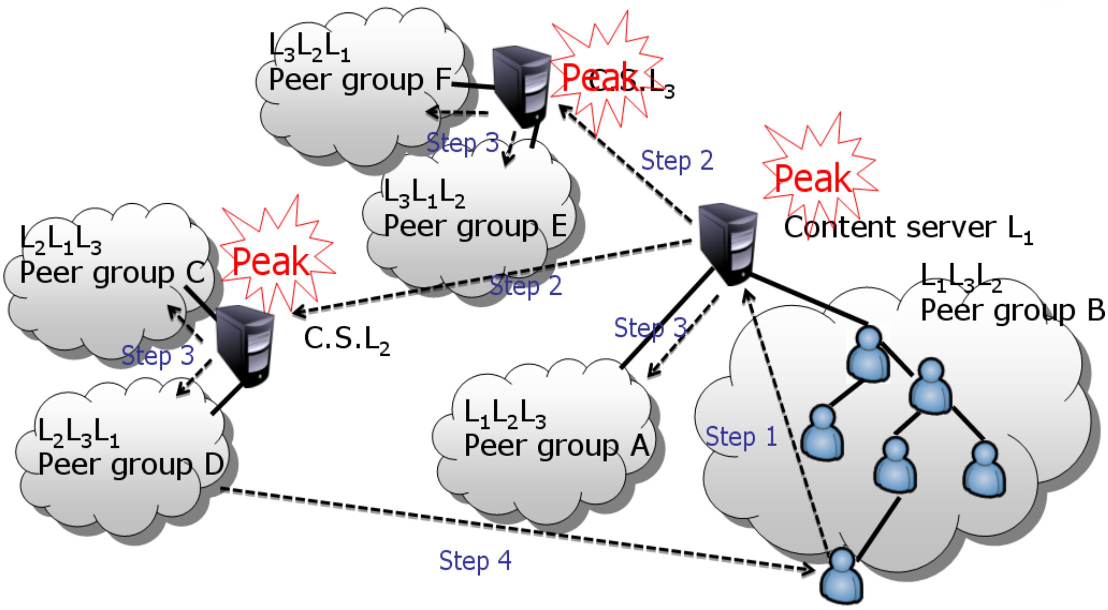


Figure 3-12：The data streaming from different group

Chapter 4

Simulation

In this section, running the prototype in the actual world via LAN with multiple users and collecting the relative true data to analysis are the best simulation. However, it is difficult and a large scale to build such a simulation environment. As FLoD is a flexible framework and already has a discrete-time simulator in P2P-NVE [27]. In order to evaluate the performance of DPN, it is a good scheme for us to adopt this simulator directly bypassing to setup the large scale simulation.

Another considerable subject in our simulation is how many groups should be created in DPN. Too many or too few groups will affect the performance of DPN obviously. The number of groups is depended on the influence of landmarks, thus the number of groups problem becomes the landmarks selecting problem. The efficiency of binning scheme using landmark [39] is based on:

(a). The type and size of topology of landmark.

(b). The connection numbers among nodes and nodes.

(c). The amount of picked landmarks.

Shieh [38] had simulated the above conditions in the previous research and got several results of his simulation. One of the results is when the number of landmarks increasing from 5 to 6, the highest RTT decrease ratio is 38%. And then the RTT decrease ratio wouldn't be improved by increasing the number of landmarks. Thus, we take this conclusion and will not simulate numbers of groups but only a few to evaluate the performance of DPN.

4.1 Simulation environment

In Figure 4-1, it is a capture of FLoD simulator [43]. This custom discrete-time simulator is coding by Microsoft Visual C++.

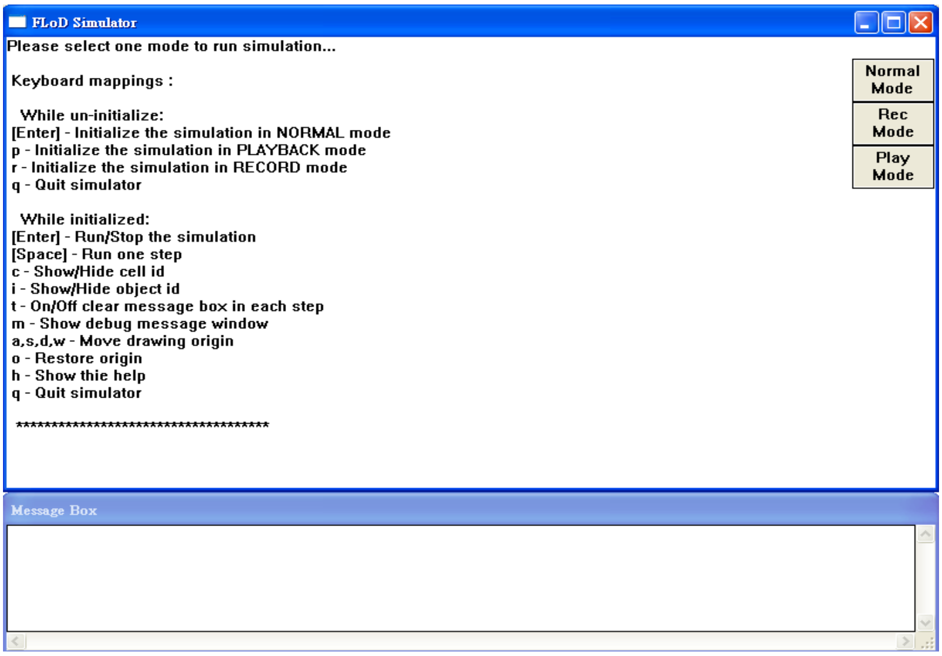


Figure 4-1：The capture of FLoD simulator [43]

The simulator runs on top of VAST, an implementation of the P2P-NVE overlay VON. To set up a P2P-NVE, a number of objects are randomly placed on a 2D map that is partitioned into square cells. For simplicity, we assume that each object has only one set of data pieces. Each node put inside the simulator to process and exchange messages with other nodes at each step. To run the simulation, a number of nodes are put randomly in the virtual environment, and stay at their joining locations until the system's average fill ratio exceeds 99%. The fill ratio is defined by the ratio of data volumes between the client's obtained data and visible data. This gives each node an initial set of data to share. The nodes then move with constant speeds, and request scene descriptions or data pieces as needed by AOI neighbors or servers if necessary. It is the capture of the running of the FLoD simulator in Figure 4-2.

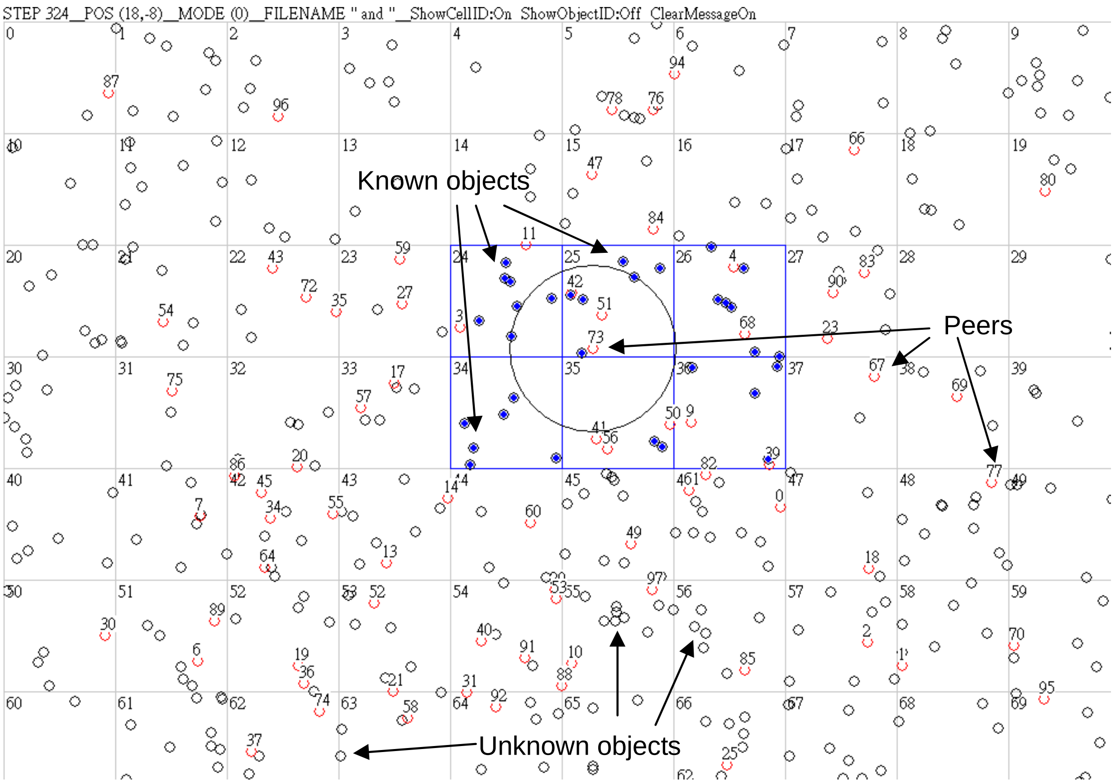


Figure 4-2：The capture of the running of FLoD simulator [43]

The fundamental setup of time-steps in this simulator is 100ms. In order to simulate the ADSL, the bandwidth limits is 1 Mbps download and 256 Kbps upload for typical broadband clients and a 10 Mbps symmetric connection for the server. Each object is set to 15 KB, where the base piece is 3 KB and 10 refinement pieces are 1.2 KB each. The scene descriptions are around 300 to 500 bytes each.

However, the key parameter, latency, between all nodes is assumed constant. To establish the situation of different latency between all nodes and to group the nodes with short latency are our task. After the above job completed, we will start all simulations proceed for 3000 steps, which is equivalent to 300 seconds while assuming 100ms per step. As we are interested in the behavior of stable condition of system, the simulation results will be based on statistics collected during last 2000 steps in each simulation. Finally, the max number of groups here is 3! = 6 due to use 3 landmarks directly. Specific simulation parameters are shown in Table 4-1.

Table 4-1：Simulation parameters

World dimension (units)	1000 x 1000
Cell Size (units)	100 x 100
AOI-radius (units)	75
Time-steps	3000
Number of nodes	50 - 500
Number of objects	500
Node speed (units / step)	1
Latency (time-steps / random)	1 - 15

4.2 Analysis

The purpose of this simulation is to evaluate the efficiency of the peers and the influence both in the peer and the server. It can be in respect to the following three metrics: transmission size, base latency and server request ratio.

4.2.1 Transmission size

Transmission size: One fundamental requirement for scalable systems is resource usage at each system, which component bounded without exceeding the component's capacity. Otherwise an overloaded component may fail or degrade its service quality. Transmission size thus is important indicators for system efficiency. In Figure4-3, we can see the performance of nodes with DPN is always higher than that of the nodes without DPN. The amount of the transmission size of DPN is better in 5 KB/m at least and 27 KB/m at most. Besides, as the nodes increasing, the higher gap of transmission size between the nodes with DPN and the nodes without DPN because the probability of content we wanted and owned by the short latency physical neighbor, which is rising no matter inside or outside the AOI.

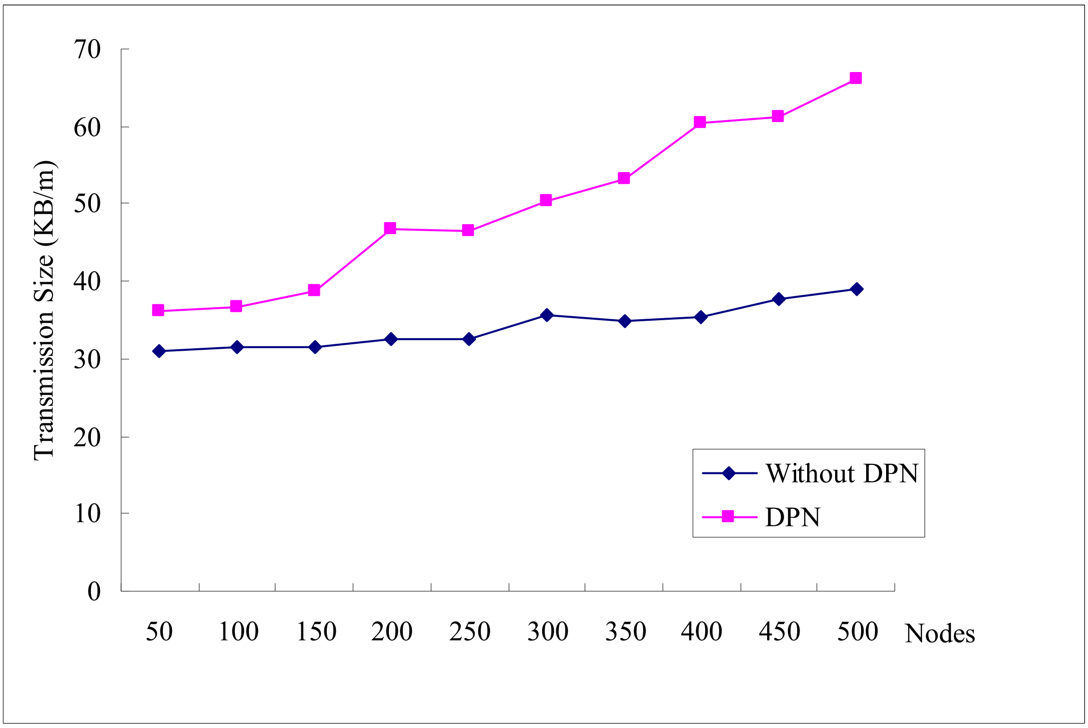


Figure 4-3：Transmission size

4.2.2 Base latency

Base latency: We define the time between the initial query and the time a base piece becomes available at a client as base latency. It serves as an indicator for how soon a user may start meaningful navigation when entering a new scene. In Figure 4-4, we can clearly feel that if the number of nodes is few, the effect of base latency is more serious to each node both with DPN and without DPN. However, the base latency of node with DPN never exceeds 2 seconds even the nodes is only 50 nodes. According to the nodes increasing, the effect will decrease to approach a constant number, yet the result of base latency without DPN is still higher than the result of base latency with DPN. Because the peer having short base latency is our first consideration, it is difficult to break by peer random choosing. It is notice that the line of optimization means the condition of latency only is 1 time-step setting among all nodes.

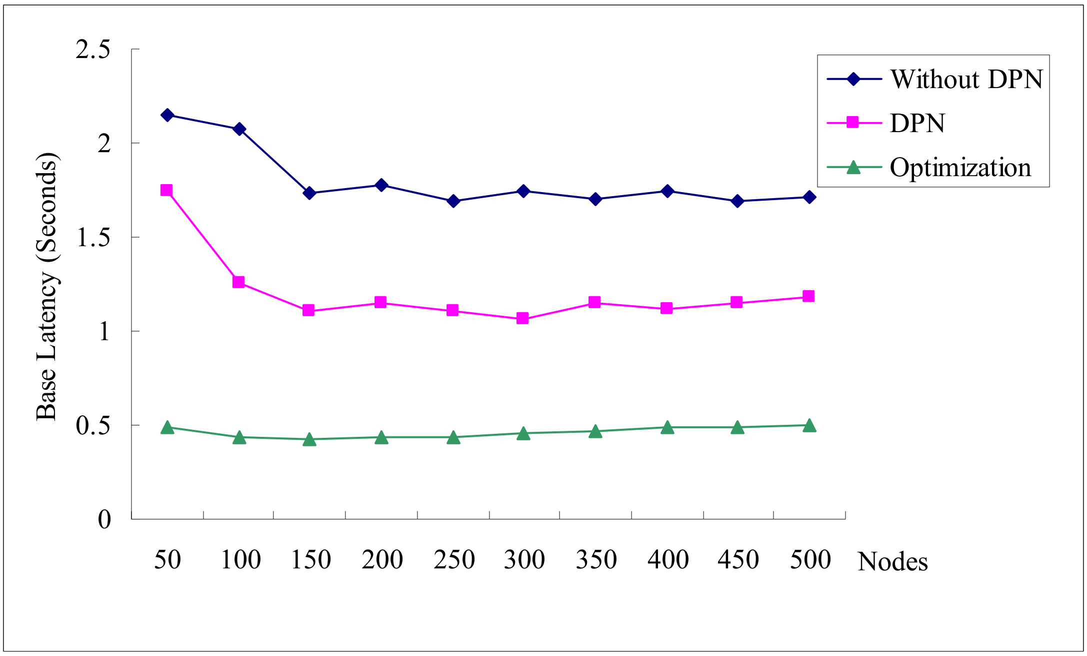


Figure 4-4：Base Latency

4.2.3 Server request ratio

Server request ratio: The loading of server is another important index. The ratio is the request counts to server divided by the all request counts. It can also refer to the scalability of P2P-NVE. On the other hand, when the nodes increasing, it means the P2P-NVE architecture is operated fine if the ratio can be always lower and keep in the constant. Illustrated in Figure 4-5 and Figure 4-6, the server request ratio including scene and object is much higher when nodes are few. However, in DPN the ratio can drops to 11% in the field of scene and 16% in the field of object at 100 nodes. That is to say, the more nodes join, the more server request ratio reduces. It denotes that if the number of the nodes is much enough, the loading of server is light no matter nodes applied DPN or not. In other words, the application of DPN is available and will not become the burden to the P2P-NVE. The simulation results show that scalability of P2P-NVE can be still performed well. It is notice that the line of optimization means the condition of latency is only 1 time-step setting among all nodes.

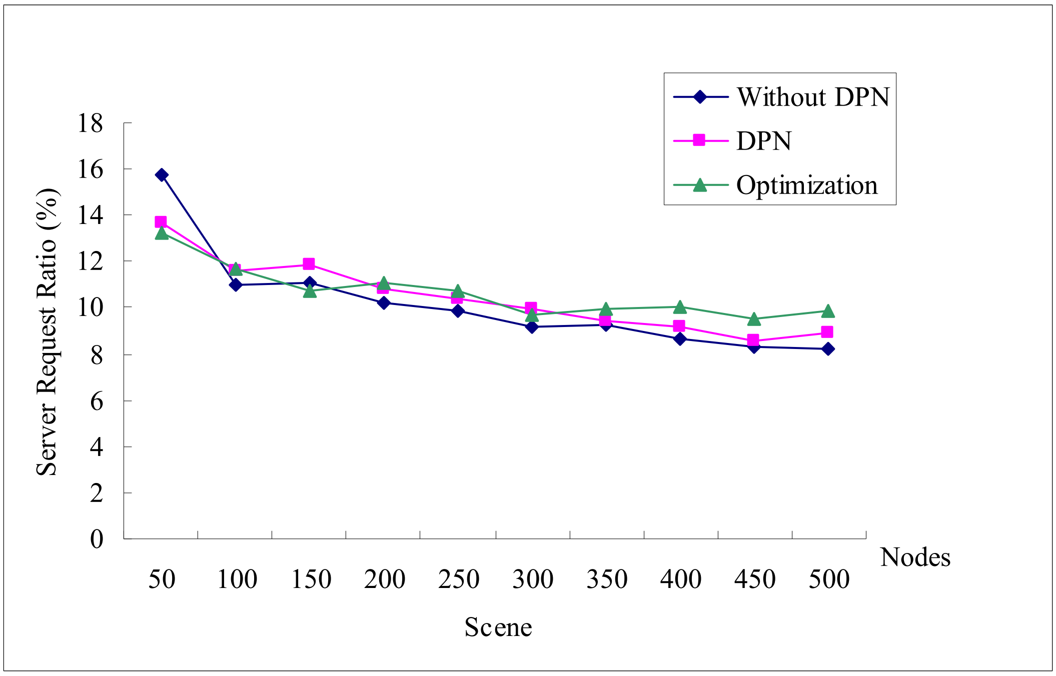


Figure 4-5：Server Request Ratio - Scene

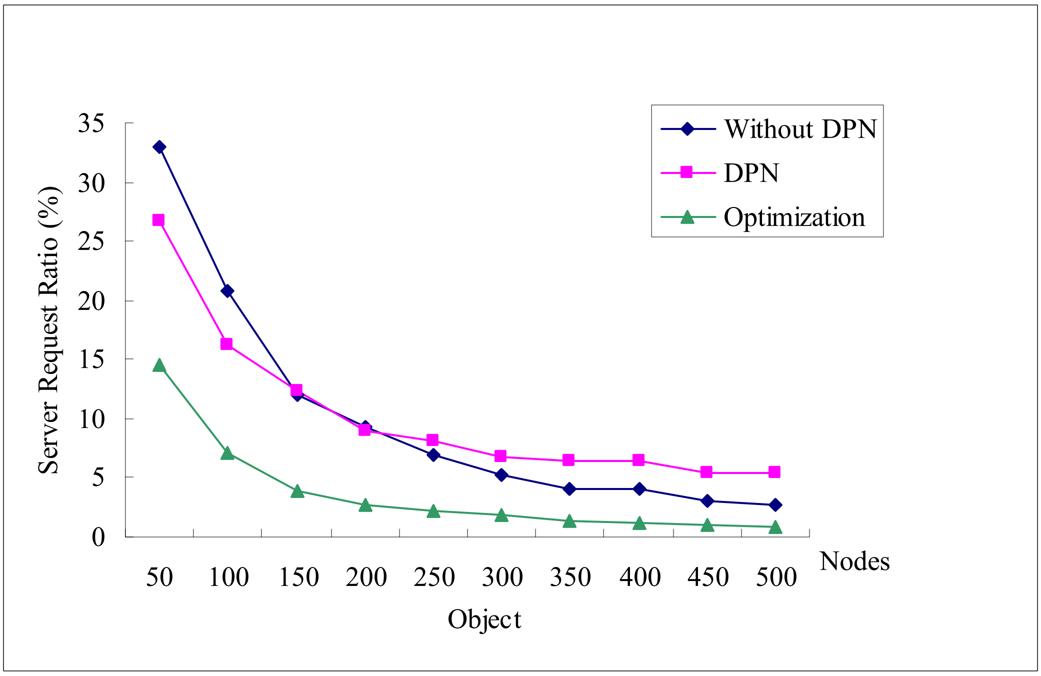


Figure 4-6：Server Request Ratio - Object

Chapter 5

Conclusion

We propose the method of approximate physical neighbor discovery called DPN in the P2P-NVE. This method is grouping the peer within the short latency by binning first. And then shares the loading of sustainability of group to each peer using base tree architecture. According to our opinion, the neighbor here we called is not only in the virtual world but also in the real world. Both of these two conditions are needed in our consideration. The advantage of this thesis is enhancing the performance of content delivery in the P2P-NVE by 45%.

However, the preservation of grouping seems not an easy thing if the number of peers is too large. The more peers join the group, the more computation is required to keep the tree architecture. Another situation is while many peers leave the P2P-NVE system in the same time, rescuing may become the heavy loading to the content server and nodes. Finally, the violent fluctuation of joining and leaving is not considered in our method. It is also a serious problem in traditional P2P architecture. We hope that these above discussed weak points can be solved by the future research.

We also analyze DPN through thoughtful simulation and get some meaningful consequences to support our stand. Consequentially, we expect to contribute a little achievement to the P2P-NVE and intend to attract further research in this developing field by more elites.

References

[1] S.inghal and M.Zyda, Neworked Virtual Environments: Design and Implementation, New York: ACM Press, 1999.

[2] J. Smed, T. Kaukoranta, and H. Hakonen, “Aspects of Networking in Multiplayer Computer Games,” Proc. ADCOG, Nov. 2001, pp. 74–81.

[3] D. C. Miller and J. A. Thorpe, “SIMNET: The Advent of Simulator Networking,” Proc. IEEE, vol. 83, no 8, Aug. 1995, pp. 1114–1123.

[4] T. Alexander, Editor, Massively Multiplayer Game Development, Charles River Media, 2003.

[5] M.Kelaskar, V. Matossian, P.Mehra, D.Paul and M.Prashar, “A Study of Discovery Mechanisms for Peer-to-Peer Applications”, Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid(CCGRID’02) PP 1, 2002.

[6] Butterfly.net, Inc. The Butterfly Grid, 2003. http:// www.butterfly.net/platform.

[7] S. Y. Hu and G. M. Liao, “Scalable Peer-to-Peer Networked Virtual Environment,” Proc. ACM SIGCOMM Wksp. on NetGames, Aug. 2004, pp. 129–133.

[8] C. Greenhalgh, J. Purbrick, and D. Snowdon, “Inside MASSIVE-3: Flexible Support for Data Consistency and World Structuring,” Proc. CVE, 2000, pp. 119–127.

[9] T. Y. Hsiao and S. M. Yuan, “Practical Middleware for Massively Multiplayer Online Games,” IEEE Internet Comp., vol. 9, no. 5, 2005, pp. 47–54.

[10] Zona Inc. Terazona: Zona application frame work, 2003.

[11] S. Y. Hu, J. F. Chen, and T. H. Chen, “Von: A scalable peer-to-peer network for virtual environments,” IEEE Network, vol. 20, no. 4, pp. 22–31, 2006.

[12] A. Goldin and C. Gotsman, “Geometric Message-Filtering Protocols for Distributed Multiagent Environments,” Presence, vol. 13, no. 3, 2004, pp. 279–95.

[13] A. Steed and C. Angus, “Supporting Scalable Peer to Peer Virtual Environments Using Frontier Sets,” Proc. IEEE Virtual Reality, Mar. 2005, pp. 27–34.

[14] B. Knutsson et al., “Peer-to-Peer Support for Massively Multiplayer Games,” Proc. INFOCOM, Mar. 2004, pp. 96–107.

[15] T. Iimura, H. Hazeyama, and Y. Kadobayashi, “Zoned Federation of Game Servers: A Peer-to-Peer Approach to Scalable Multi-Player Online Games,” Proc. ACM SIGCOMM Wksp. NetGames, Aug. 2004, pp. 116–120.

[16] Y. Kawahara, T. Aoyama, and H. Morikawa, “A Peer-to-Peer Message

Exchange Scheme for Large-Scale Networked Virtual Environments," Telecomm. Sys., vol. 25, no. 3–4, 2004, pp. 353–370.
[17] J. F. Chen et al., "A Message Interchange Protocol Based on Routing Information Protocol in a Virtual World," Proc. AINA'05, Mar. 2005, pp. 377–384.
[18] J. Keller and G. Simon, "Towards a Peer-to-Peer Shared Virtual Reality," Proc. 22nd ICDCS (Wksps.), July 2002, pp. 695–700, http://solipsis.netofpeers.net
[19] S. Y. Hu, "Scalable Peer-to-Peer Networked Virtual Environment," Master's thesis, Tamkang Univ., Taiwan, Jan. 2005.
[20] S. Rooney, D. Bauer, and R. Deydier, "A Federated Peer-to-Peer Network Game Architecture," IEEE Commun. Mag., vol. 42, no. 5, 2004, pp. 114–122.
[21] A. Yu and S. T. Vuong, "MOPAR: a Mobile Peer-to-Peer Overlay Architecture for Interest Management of Massively Multiplayer Online Games," Proc. NOSSDAV, Jun. 2005, pp. 99–104.
[22] F. Aurenhammer, "Voronoi Diagrams: A Survey of a Fundamental Geometric Data Structure," ACM Comp. Surveys, vol. 23, no. 3, 1991, pp. 345–405.
[23] Y. Cui, B. Li, and K. Nahrstedt, "ostream: asynchronous streaming multicast in application-layer overlay networks," IEEE JSAC, vol. 22, no. 1, pp. 91–106, 2004.
[24] N. Magharei and R. Rejaie, "Prime: Peer-to-peer receiver-driven meshbased streaming," in Proc. INFOCOM, 2007, pp. 1415–1423.
[25] E. Teler and D. Lischinski, "Streaming of complex 3d scenes for remote walkthroughs," CGF (EG 2001), vol. 20, no. 3, 2001.
[26] S. Y. Hu, "A case for 3d streaming on peer-to-peer networks," in Proc. Web3D, 2006, pp. 57–63.
[27] S. Y. Hu, T. H. Huang, S. C. Chang, W. L. Sung, J. R. Jiang, and B.Y. Chen, "FLoD: A Framework for Peer-to-Peer 3D Streaming," in Proc. IEEE INFOCOM, Apr. 2008.
[28] J. Chim, R. W. H. Lau, H. V. Leong, and A. Si, "Cyberwalk: A web based distributed virtual walkthrough environment," in IEEE TMM, vol. 5, no. 4, pp. 503–515, 2003.
[29] Y. Liu, L. Xiao, X. Liu, L.M. Ni, and X. Zhang, "Location Awareness in Unstructured Peer-to-Peer Systems," in IEEE Trans. Parallel and Distributed Systems, vol. 16, pp. 163-174, 2005.
[30] Y. Liu, Z. Zhuang, L. Xiao, and L.M. Ni, "A Distributed Approach to Solving Overlay Mismatch Problem," in Proc. 24th Int'l Conf. Distributed Computing Systems (ICDCS '04), 2004.
[31] Y. Liu, L. Xiao, X. Liu, L.M. Ni, and X. Zhang, "Building a Scalable Bipartite

P2P Overlay Network," in IEEE Trans. Parallel and Distributed Systems, vol. 18, pp. 1296-1306, 2007.

[32] D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin und R. Panigrapy, "Consistent hashing and random trees: Distributed cashing protocols for relieving hot spots on the World Wide Web", in Proc. STOC, pp.654-663, 1997

[33] 劉之揚, "使用熱門廣播法及支援點對點傳輸之影音內容傳遞網路", 國立中央大學資訊工程研究所碩士論文, 1993年

[34] James D. Guyton and Michael F. Schwartz. Locating nearby copies of replicated internet servers. In Proceedings of SIGCOMM '95, August 1995.

[35] Steven Michael Hotz. "Routing information organization to support scalable interdomain routing with heterogeneous path requirement," PhD thesis, Univ. of S. California, U.S.A, 1994.

[36] Hirokazu Miura and Miki Yamamoto. " Content Routing with Network Support Using Passive Measurement in Content Distribution Networks", in Proceedings of the IEEE International Conference on Computer Communications and Networks, Miami, FL , October 2002.

[37] Ng, TSE, and H. Zhang. "Towards Global Network Positioning." Proceedings of ACM SIGCOMM IMW 2001, San Fransisco, CA, USA, pp. 25-35, Nov. 2001.

[38] 謝民一, "複製可追查型CDN-P2P遞送網路下之近端伺服器選擇", 國立中央大學資訊工程研究所碩士論文, 1994年

[39] S. Ratnasamy, M. Handley, R. Karp and S. Shenker, "Topologically-aware overlay construction and server selection", in Proc. INFOCOM, 2002, pp. 1190-1199.

[40] H. Hoppe, "Progressive meshes," in Proc. SIGGRAPH, 1996.

[41] E. W. Zegura, K. Calvert and S. Bhattacharjee, "How to model an Internetwork", in *Proc. IEEE INFOCOM*, San Francisco, CA, 1996, pp. 594-602.

[42] Gao Huang, Meng Ye, Long Cheng, "Modeling System Performance in MMORPG", in IEEE CSG Workshops, 2004, pp. 512-518.

[43] The Simulator. http://ascend.sourceforge.net/