國　立　中　央　大　學

資 訊 工 程 研 究 所

碩　士　論　文

Efficient AOI-Cast for Peer-to-Peer Networked Virtual Environments

同儕式網路虛擬環境高效率互動範圍群播

研 究 生：黃宇立

指導教授：江振瑞 博士

中 華 民 國 九 十 六 年 七 月

Efficient AOI-Cast for Peer-to-Peer Networked Virtual Environments

Student：Yu-Li Huang

Advisor：Dr. Jehn-Ruey Jiang

A Thesis

Submitted to Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Central University

in Partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science and Information Engineering

July 2007

Jhongli City, Taiwan, Republic of China

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)，爲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)

(V)同意 (立即開放)
(　)同意 (一年後開放)，原因是：________________
(　)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: 黃　宇　立　　學號： 945202052

論文名稱: Efficient AOI-Cast for Peer-to-Peer Networked Virtual Environments

指導教授姓名： 江　振　瑞

系所　： 資訊工程 所　□博士班　■碩士班

日期：民國 96 年 7 月 11 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。

論文名稱：同儕式網路虛擬環境高效能互動範圍群播

校所組別：國立中央大學 資訊工程研究所　　　　　　　　　頁數：27

研究生：黃宇立　　　　　　　　　　　　　　　　指導教授：江振瑞　博士

中文摘要：

網路虛擬環境(Network Virtual Environment – NVE)是一個由電腦產生的虛擬世界，它可以讓使用者們透過網路連線，經由訊息交換來與其他使用者們互動。每位使用者在網路虛擬環境中都只注意一個有限的範圍，並只與在此範圍內的使用者們做互動，我們稱此範圍為一個互動範圍(Area of interest – AOI)。網路虛擬環境通常都需要支援許多使用者同時使用，所以提升其擴充性(scalability)是很重要的。由於擴充性的問題也就是資源上的問題，要提升擴充性，就必須要減少資源消耗，我們將著重在減少頻寬消耗方面。目前一些同儕式方法已經被提出來用以改善系統的擴充性，但是關於互動範圍內的擴充性卻仍未有方法來提升它。我們提出的兩個方法 VoroCast 和 FiboCast，希望能夠改善互動範圍的擴充性。其中 VoroCast 讓訊息的傳輸沒有多餘的封包以及可以應用彙整(aggregation)和壓縮(compression)的機制來減少頻寬消耗。而 FiboCast 則是 VoroCast 的一個改進，通常是應用於使用者處於擁擠的環境下，靠著調整訊息散佈的範圍來更進一步地減少頻寬消耗。我們利用模擬實驗來衡量我們所提出的方法的效能。實驗結果證明了我們的方法的確可以達到較少的頻寬消耗，而且還保有很高的鄰居關係的一致性(neighborship consistency)及較少的偏移距離(drift distance)。

關鍵字：同儕式計算，網路虛擬環境，互動範圍群播，擴充性。

Efficient AOI-Cast for Peer-to-Peer Network Virtual Environments

Student：Yu-Li Huang Advisor：Prof. Jehn-Ruey Jiang

Department of Computer Science and Information Engineering,

National Central University

Jhongli City, Taoyuan, 320, Taiwan

Abstract —A networked virtual environment (NVE) is computer-generated virtual world where users interact with each other by exchanging messages via network connections. Each user in NVE only pays attention to a bounded area called *area of interest* (AOI). NVE usually has a lot of users, so it is important to higher the scalability. Several peer-to-peer (P2P) schemes are proposed to improve the system scalability which indicates the ability to handle growing amounts of users in the system. However, they do not consider the AOI scalability which indicates the ability to handle growing amounts of users within AOI. We proposed VoroCast and FiboCast to improve the AOI scalability. VoroCast makes the message transmission non-redundant and applies aggregation and compression mechanisms to reduce the bandwidth consumption. FiboCast is the improvement of VoroCast. It is used especially for a user who is in a crowded environment and further reduces the bandwidth consumption by adjusting the message dissemination range. We perform simulation experiments to evaluate the performances of VoroCast and FiboCast.

Keywords —Peer-to-Peer, NVE, Voronoi Diagram, Area of Interest, Scalability.

Contents

List of Figures

1 Introduction

A networked virtual environment (NVE) is computer-generated virtual world where multiple geographically distributed users can assume virtual representatives (or avatars) to interact with each other by exchanging messages via network connections. Each user in NVEs is seen as a coordinate point on a 2D plane and only pays attention to a bounded area called the *area of interest* (AOI). The AOI of a user u is usually defined to be a circular area centered at u; all users within u's AOI is called u's AOI neighbors. A user has to be aware of all its AOI neighbors. Therefore, a user has to send message about its state changes to the users whose AOIs include itself. If we assume an equal-sized AOI for every user as most NVEs do, then each user has to send messages to and receive messages from all its AOI neighbors. In this thesis, we use the term AOI-cast to refer to the message transmission from a user to all its AOI neighbors, and we pursuit efficient AOI-cast scheme.

NVEs usually have a lot of users. For example, a massively multiplayer online game (MMOG), a special case of NVEs, has to support hundreds of thousands of players (users) simultaneously in the virtual world. Therefore, how to make the NVE scalable is an important issue. We would like the NVE to have better system scalability and AOI scalability. The system scalability of an NVE indicates its ability to handle growing amounts of users in the system; the AOI scalability of an NVE indicates its ability to handle growing amounts of users within AOI.

The server-based architecture is a common architecture for NVEs today. Since a server (or server cluster) has limited resources and all loads are concentrated on the server, the system scalability of server-based architecture is low. Several schemes, such as SimMUD [11], VON [6], P2P-MES [8], Solipsis [9, 10], etc., are proposed to improve the system scalability by using the peer-to-peer (P2P) architecture. The P2P architecture has higher system scalability than server-based one because it has the advantage of distributing loads to all user computers (peers or nodes). Peers are not only resource consumers but also resource providers. When more peers join the NVE,

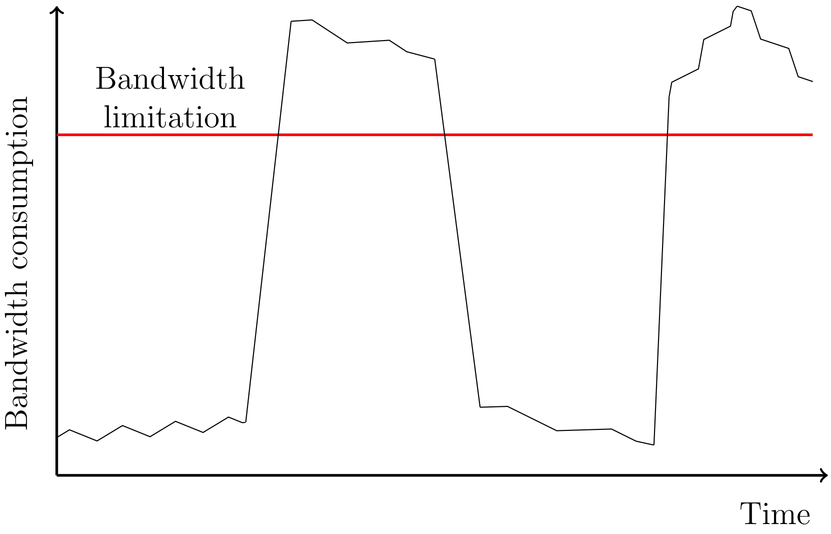

Figure 1: The peak bandwidth consumption of a node.

there are more resources provided. Since the P2P-based NVE has higher scalability, we focus it in this thesis.

All P2P-based NVEs support AOI-cast. We categorize the AOI-cast schemes into two classes: directly sending schemes and forwarding schemes. In directly sending schemes, such as Solipsis [9, 10], P2P-MES [8] and VON [6], each peer connects to all AOI neighbors and sends message to them directly. The advantage of the directly sending scheme is low transmission latency, but it has the drawback of large peak bandwidth consumption (as Figure 1). When there are many peers in a peer's AOI, the peak bandwidth consumption may exceed the bandwidth limitation, causing negative influence on the AOI scalability. In the forwarding schemes, such as SimMUD [11], APOLO [7] and VON-Forwarding [5], each peer sends the message to the some neighbor peers, which in turn relay the message to other neighbor peers until the message is received by all AOI neighbors. Although forwarding schemes have longer transmission latency than directly sending schemes, they have better AOI scalability by distributing the bandwidth consumption among AOI neighbors.

In this thesis, we propose two forwarding AOI-cast schemes, VoroCast and FiboCast, to improve the AOI scalability for P2P NVEs by (1) eliminating redundant messages and (2) by allowing nodes dynamically adjust the message dissemination range.

VoroCast divides AOI neighbors by the Voronoi diagram [3] and each peer only connects with some nearby neighbors (enclosing neighbors). VoroCast has the advantage that its message transmission is non-redundant and the bandwidth consumption can be further reduced by applying aggregation and compression mechanisms. FiboCast is useful for the AOI-cast of essential messages that have to be sent periodically in NVEs, such as position update, etc. since a user usually pays more attention to near events than to far events, FiboCast dynamically adjusts the message dissemination range according to Fibonacci sequence so that the neighbors nearer the AOI center get messages more frequently, while the farther neighbors get messages less frequently. Consequently, FiboCast can achieve higher AOI scalability than VoroCast. We perform simulation experiments to evaluate the performances of VoroCast and FiboCast; we also compare them with related schemes.

The rest of the thesis is organized as follows. We describe some research work related to P2P AOI-cast in chapter 2. In chapter 3, we present the design of VoroCast and FiboCast. In chapter 4, we evaluate VoroCast and FiboCast by simulations and make some discussions. Finally, we draw a conclusion in chapter 5.

2 Related work

Several schemes are proposed for P2P NVEs, such as Solipsis [9, 10], P2P-MES [8], VON [6], SIMMER [11], provide schemes for AOI-cast. According to their multicast mechanisms, we categorize the AOI schemes into two classes: directly sending schemes and forwarding schemes. Below, we briefly describe some schemes of the two classes.

2.1 The directly sending schemes

The Solipsis [9, 10] presents a virtual world and intends to be scalable for a large number (million, billion or more) of participants. In Solipsis, each node directly connects to the other nodes within its awareness area and does neighbor discovery by mutual notification. In order to ensure the mutual notification, each node should be covered in a convex hull (i.e., ensures the Global Connectivity property) that is composed by some nodes within awareness area. If the node is not covered by a convex hull (as Figure 2(a)), the node has to recover the convex hull by finding the nodes that can form a new one (as Figure 2(b)). However, the recovery procedure delays the neighbor discovery, the topology becomes inconsistent.

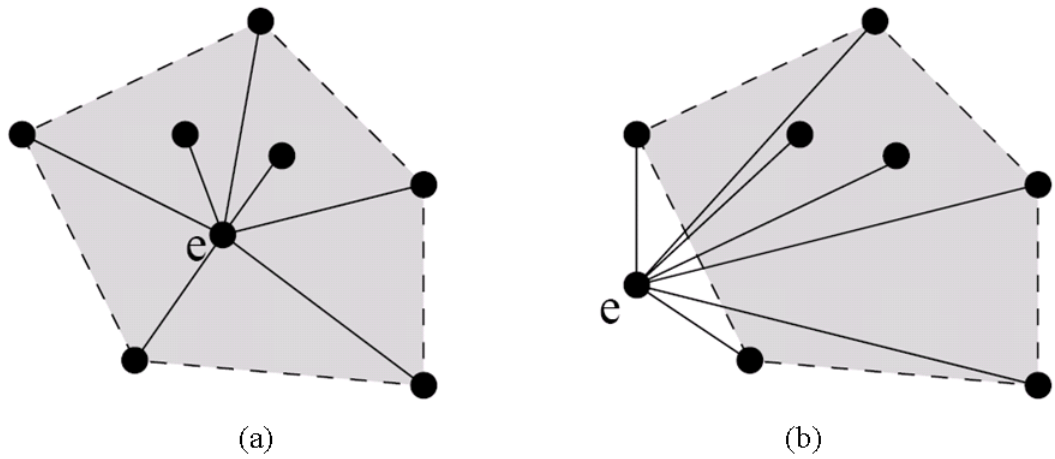

Figure 2: (a) The node e is covered by a convex hull; (b) The node e is not covered by a convex hull and it has to be recovered.

Kawakawa et al. [8] proposed a message exchange scheme to discover neighbors. In their proposal, each node takes the nearest n entities which are the active entities (AE) with direct connections. Except the nearest n entities, the other entities are latent entities (LE). As Figure 3 shows, each entity compresses its active entities' rough information into list and to exchange the list with its active entities. The information of latent entity is known by the exchange mechanism. MES has an advantage that the bandwidth use for neighbor discovery is constant, because of the number of active entity is predefined by the system. Due to the bounded usage of bandwidth, the system can scale well, but its AOI is restricted by the number of active entity. Besides, MES has another serious problem that the overlay partition may be occurred when the predefined number of active entity is low and the node distribution is non-uniform.

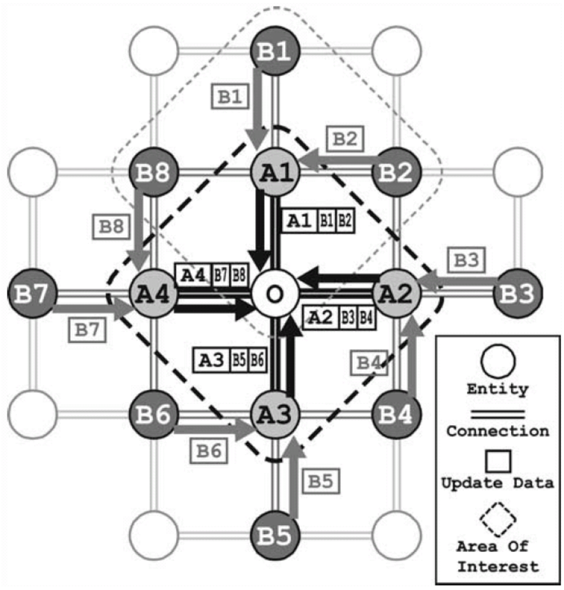

Figure 3: The message exchange scenario of P2P-MES.

In VON [6], each node has an AOI range and divides the AOI neighbors by Voronoi diagram (see as Figure 4). Each node connects to AOI neighbors by directed connection. When a node moves, the node sends its position update to all connected neighbors. If the recipient is a boundary node (i.e., the node whose enclosing neighbors may partially lie outside the AOI, the stars and triangles in Figure 4), the boundary node compares the difference between its enclosing neighbors and the knowledge of the moving node, then the boundary node sends back the up to date enclosing neighbor information to moved node. Each node discovers its neighbors by comparing with its boundary neighbors recursively. Because VON connects to all AOI neighbors directly, the AOI of a node can not accommodate too many nodes. If the number of AOI neighbors of a given node exceeds a threshold, the node shrinks its AOI radius to reduce the number of AOI neighbors. Although VON scheme performs well consistency and low latency, it consumes more bandwidth on connection and message transmission.

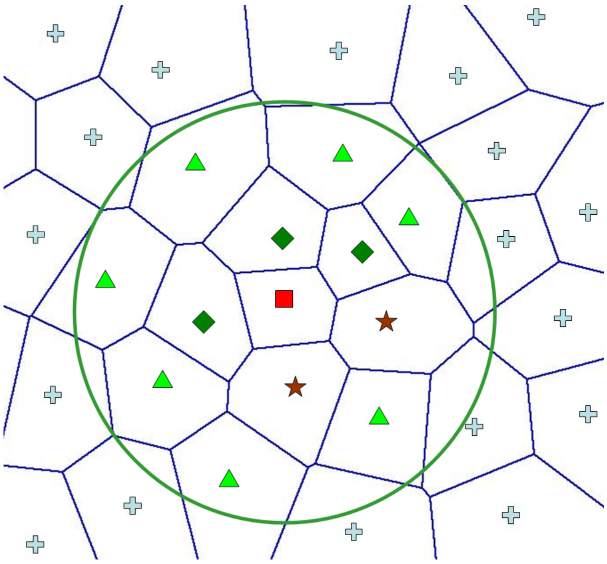

Figure 4: Voronoi diagram: the large circle is the AOI boundary for the center node; square is the center node; diamonds are enclosing neighbors; triangles are boundary neighbors; stars are both enclosing neighbors and boundary neighbors; circles represents a regular AOI neighbors; crosses represent neighbors irrelevant (i.e., outside of AOI) to the center node.

2.2 The forwarding schemes

SimMud [11] built the game by Pastry [12] and Scribe [4] as its P2P infrastructure. Pastry is a widely used P2P overlay and Scribe is the multicast infrastructure built on top of Pastry. The game world is divided into regions based on the limited sensing capabilities of a player's avatar. Each region has an ID and the node whose ID is the closest to the region ID as the coordinator. The coordinator not only coordinates all shared objects in the region, but also serves as the root of the multicast tree. As Figure 5 shows, players in the same region form an interest group and the state update messages are only disseminated within the group. If a player wants to send a message to certain destination player, the message will be sent to the coordinator first. Then the coordinator sends the message to the destination player by the multicast tree. However, the single multicast tree is not flexible for message transmission, because the single multicast tree makes the message has to relay more hops even the two players are adjacent. Additionally, it is not flexible that the regions in SimMud are fixed and restricts the events must in the same region. In other words, there is no interaction between regions.

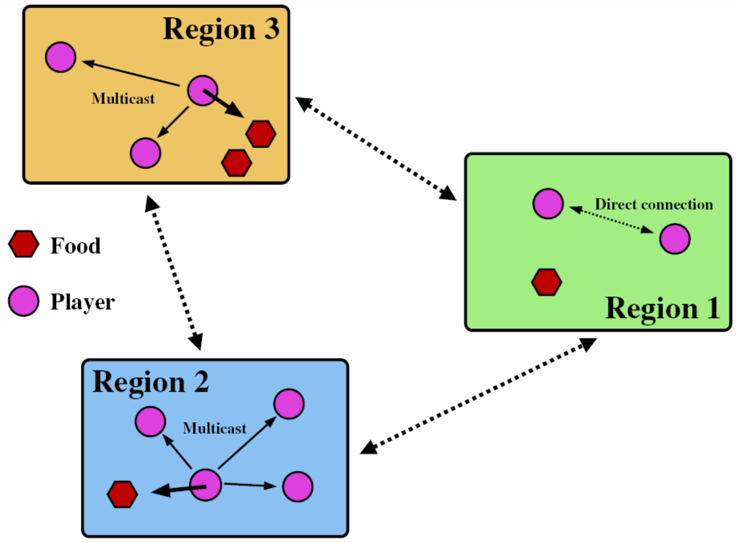

Figure 5: The game design of SimMud.

APOLO [7] proposes a protocol that distributed constructs an overlay. In APOLO, each node uses two hop beaconing protocol to maintain four out-direction links to the nearest neighbors in a two-dimensional plane (as Figure 6. Because each quadrant has an out-direction link, we regard the node partition as quadrant-based partition. Also, each node has the knowledge about which nodes are linking to itself with out-direction. These links are the in-direction links. For the connectivity of the network topology, there are four special-purpose virtual nodes, namely Portals. The four portals are located in the corners of the topology and the some border nodes can direct to the portal without disconnecting. The objective of APOLO is that each node only has to manage a small number of links. When a node multicasts a message, the recipient node forwards the message according the following rule. If the root node and the recipient node are both position in the same quadrant of the next hop and there is an in-direction link between the recipient node and the next hop. The non-redundant message transmission on APOLO due to the restricted direction links. However, the restricted direction links also make the message transmission inefficient. A node can not forward the message to the nearby node even if there is a link between the two nodes. As Figure 6 shows, each node connects to other nodes with directed links. When node s4 multicasts a message, the message to node s8 is pass through the node s6, s3, s7. Although there is a link between node s6 and s8, the message can not relay on the link due to the restricted direction.

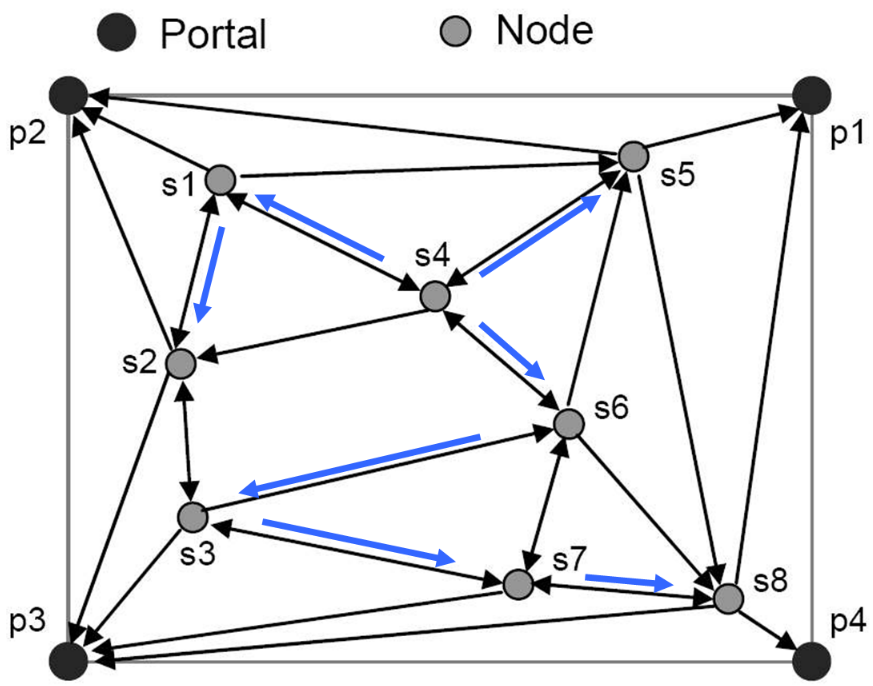

Figure 6: The network topology of APOLO. The blue arrows are the message forwarding paths of node s4.

VON-Forwarding model [5] is an extension of VON, it proposed that each node only connects with enclosing neighbors instead of connecting with all AOI neighbors. The message forwards to destination node through relaying by enclosing neighbors. The method not only takes approximately constant number of connection, but also can use aggregation and compression to reduce the bandwidth consumption. Thus, VON-Forwarding model can accommodate more nodes within AOI than VON. VON-

Forwarding model, unlike APOLO, is able to forward messages without the directed restriction on links, the message can be sent to destination as soon as possible. But, as Figure 7 shows, without the restriction causes the transmission has redundant messages. So, if the redundant messages could be eliminated from transmission, Voronoi-based partition should more efficient than quadrant-based partition.

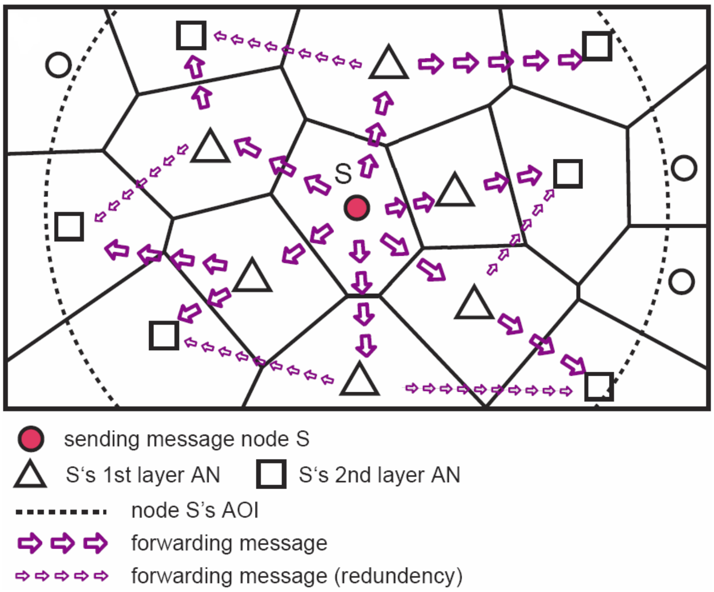

Figure 7: The forwarding path on VON-Forwarding model.

2.3 Comparisons of directly sending schemes and forwarding schemes

The directly sending schemes all have the advantage of low latency because the message are sent to any neighbor nodes directly. However, they also have a common problem on peak bandwidth consumption. Because the bandwidth of a node is limited, if the node has to send message to too many neighbors at the same time, it will exceed the bandwidth limits. When the situation occurs, it affects the system performance seriously. Even the bandwidth usage of a node is not exceeding the limits, maintaining so many connections are also an overhead. On the contrary, the forwarding schemes only connect with few nodes. The message from source node to destination node relays by the connected nodes, it distributes the bandwidth consumption to other nodes. So the forwarding transmission consumes less bandwidth. However, the relaying mechanism causes the higher latency than directed sending, but the forwarding schemes can send to more destinations than directly sending schemes in the same period time. For example, supposing each point to point transmission needs 100 ms and each node can handle 10 transmissions. In the period of 200 ms, the node can transmit the message to 20 destinations by direct sending, but the node can transmit the message to 110 destinations. Although the example is a little exaggerated, it indicates the forwarding scheme is more efficient on transmission. In this thesis, we only care about the AOI-cast, so we compare the directly sending AOI-cast and forwarding AOI-cast in Table 1.

Table 1: The comparison of directly sending AOI-cast and forwarding AOI-cast

	Directly sending AOI-cast	Forwarding AOI-cast
Connection	All AOI neighbors	Selected few
Latency	Low	Depends on the forwarding hops
Bandwidth consumption (at the source node)	High	Low
Aggregation	Not suitable	Suitable
Compression	Not suitable	Suitable
Topology	Star	Tree

3 Proposed schemes

In this chapter, we proposed VoroCast and FiboCast to improve the AOI scalability by reducing the bandwidth consumption. We describe the details of the schemes in next two sections, respectively.

3.1 VoroCast

3.1.1 Details of VoroCast

In VoroCast, no node receives redundant messages. The basic idea is to construct a multicast tree spanning all the AOI neighbors. The messages are transmitted along the branches so that there is no redundant message. Each node in VoroCast selects the forwarding nodes by local information to construct a spanning tree.

Each node in VoroCast has a unique ID and is represented as a point in the Voronoi diagram. Each node x has an associated Voronoi regions and two kinds of special AOI neighbors. One is *enclosing neighbors* (ENs) also called one-hop neighbors, whose regions directly surround x's region. The other is *two-hop neighbors*, which are one-hop neighbors of one-hop neighbors. By Voronoi diagrams, a node x can obtain one-hop neighbor list. And by exchanging one-hop neighbors list with each one-hop neighbors, the information of one-hop and two-hop neighbors. For a node, we would like to construct a tree rooted a r and spanning all r's AOI neighbors. We have the following two rules for a node x:

- If x is the root node (i.e., the node r initiating the AOI-cast), it transmits the message to all of its one-hop neighbors.
- If x is not the root node (i.e., an intermediate node or a leaf node), it executes a child node selection procedure to select its children to forward the message.

All of the nodes obey the rules to relay the message until the message reaches all neighbors. Below, we describe the derails of the child node selection procedure.

In children selection procedure, we would like all nodes, except the root node, have only one parent node. If a node has only one parent node, it means there is only one unique path from the root node to the node. Therefore, there is no redundant message transmitted. Figure 8 is the pseudo code of the child node selection procedure, where dist (a, b) stands for the Euclidean distance between a and b.

```
//for node x to select child nodes to forward a message m
if (x == root_node)
  send m to all enclosing neighbors
else
  for each enclosing neighbor y of node x, y ≠ parent of x
    if (y is in the AOI range of root_node)
      parent = min (<dist (z, root_node), z>), z is enclosing neighbors of y, y ≠ x
      if (parent == x)
        send m to y
```

Figure 8: The pseudo code of the child node selection procedure.

In this pseudo code, the <dist (z, root_node), z> is a ordered pair. The function dist (z, root_node) will return the Euclidean distance between z and root_node. An ordered pair(a, b) precedes another pair(c, d) if a<c or a≤c and b<d. By the pseudo code, node x checks for each x's enclosing neighbor y, except x's parent node, from which x receives the message, if x is the parent node of y. If so, the x sends the message to y. Node y takes x as its parent node among all its enclosing neighbors if x has the minimum Euclidean distance to the root node.

After the child node selection procedure, a node forwards the received messages to all its child nodes. If the messages are sent to the same destination, they will be aggregated to share a common header. The aggregated message can also be compressed. The two mechanisms can further reduce bandwidth consumption.

3.1.2 An example of VoroCast

We give an example to explain how the scheme works. Figure 9 shows the message forwarding path of root node. Each node in network has a unique ID (i.e., the number in the parentheses) and maintains the information of two-hop neighbors. The solid line represents the message actual forward path which is selected by VoroCast. The dot line represents the candidate path before selection. The number is located aside the line is the distance to root node. When the root node disseminates its position update message, the message includes the position and AOI range of the root node. The message transmits to all of the enclosing neighbors of the root node (i.e., node A, B, C, D and E) first. After node A, B, C, D and E get the message from root node, they have to relay this message to their enclosing neighbors which are located at root node's AOI. We focus on node A to illustrate the progress of VoroCast. The candidate set of node A is composed of root node, node B, E, F and G. Then node A uses the viewpoint of the four nodes except its parent, root node, to make the decision.

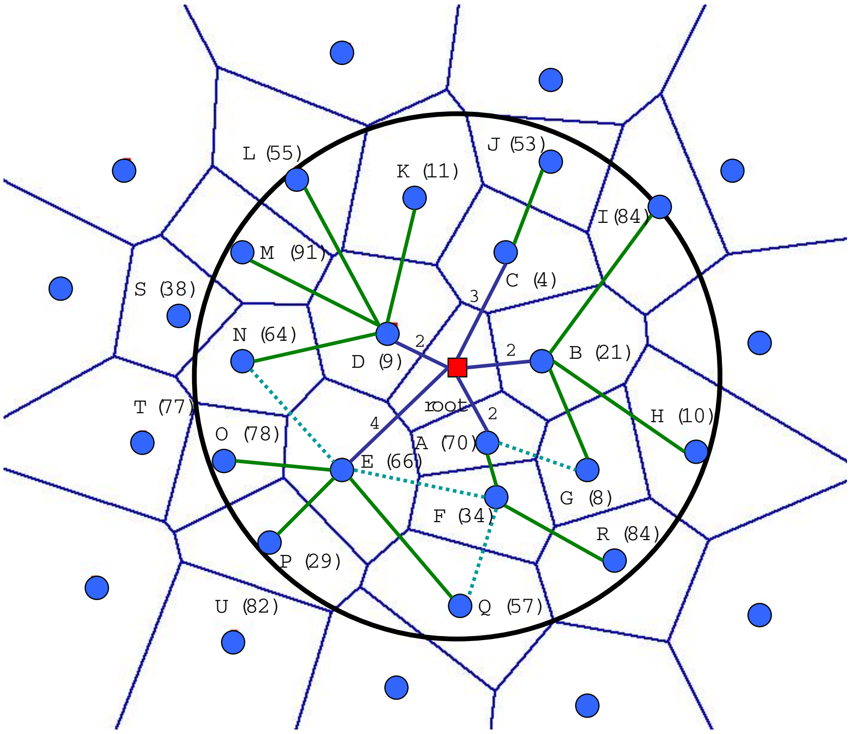

Figure 9: The message forwarding path of the root node.

Node B discovers that its enclosing node with the shortest distance to root node is root node. Node B thinks that root node is its parent and not node A, so node A will not forward the message to node B.

Node E does the same process as node B.

Node F discovers that node A has the shortest distance to root node among its enclosing neighbors. So node A is the parent of node F, and node A will forward the message to node F.

Node G discovers that node A and node B have the shortest distance to root node at the same time. Therefore, node G compares the ID of node A and B to select a unique parent. Node B has the smallest ID and become the parent of node G at last. So node A is not the parent of node G and will not relay the message to node G.

The other nodes base on the above rules and select their next forwarding nodes until the message reaches the AOI border of the root node. And the total path would be a spanning tree.

3.2 FiboCast

3.2.1 Details of FiboCast

In NVEs, some messages have to be sent periodically, such as position updates, etc. Generally speaking, users usually more care about the surrounding events than the farther events. FiboCast dynamically adjusts the message dissemination range measured by hop count according the Fibonacci sequence so that the neighbors near the AOI center get message more frequently, while farther neighbors get messages less frequently. Fibonacci sequence is a sequence of numbers, in which a number is the sum of two previous numbers. By setting the first two numbers in the sequence, we have different Fibonacci sequences.

We use hop count to adjust the message dissemination range in FiboCast because the hop can reflect the crowded situations of portions of AOI. For example, as shown in Figure 10, the left portion of te AOI is morecrowded than the right portion. The nodes in the left portion thus is more hops away from the centered node than the nodes in the right portion. The largest hop count of the transmission is based on the Fibonacci sequence. We would like the sequence increases slowly while the number of the sequence is small. As the number of the sequence is more and more bigger, it can increases quickly. The messages can be sent to the nearer neighbors more frequently and to the farther neighbors less frequently by this type of sequence. We have considered some sequences, such as exponential sequence, linear sequence, etc. However, these sequences can not reach our expectation. For example, the linear sequence can not increase quickly as the number of sequence is getting bigger so that the farther neighbors still can get messages frequently; the incremental rate of exponential sequence is too high so that the near neighbors get messages with too low frequency. Therefore, we adopt Fibonacci sequence to adjust the message dissemination range.

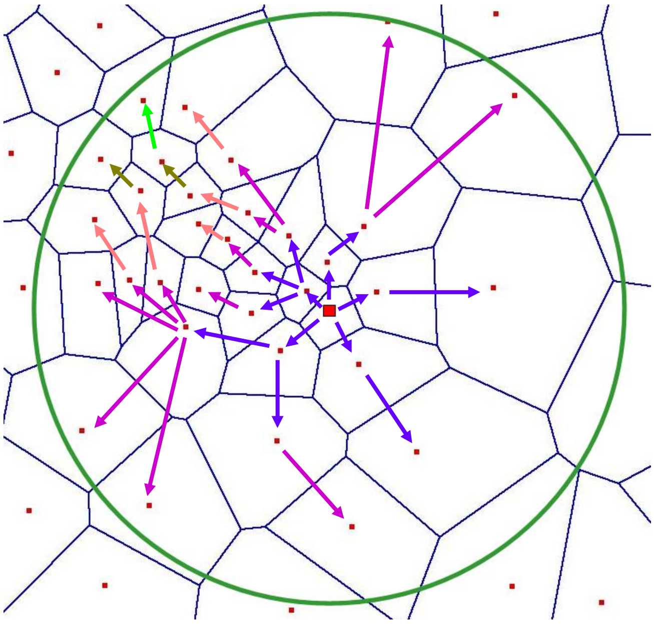

Figure 10: The example of a partial crowded situation.

There are several variables used in FiboCast. The current hop (*curr_hop*) varies between the prespecified minimum hop (*min_hop*) and maximum hop (*max_hop*), and the *fib_num* is derived sequentially from a Fibonacci sequence. The *min_hop* can be two or more for keeping the child node selection procedure works normally. The *max_hop* indicates the maximum number of hops that a message can go through. As we will show later, it is adjusted periodically. The *curr_hop* affects the message dissemination range of AOI-cast and it is defined to be

$$curr\_hop = \begin{cases} min\_hop + fib\_num, & \text{if } curr\_hop \leq max\_hop \\ 0, & \text{if } curr\_hop > max\_hop \end{cases}$$

Table 2: The example of FiboCast. V represents that the message will be forwarded to this hop; X represents the message will not be forwarded to this hop; CH stands for the *curr_hop*; FN stands for the current Fibonacci number; Reduced is the number of the saved message at current step.

Steps \ hops	1	2	3	4	5	6	7	8	CH	FN	Reduced
1	V	V	X	X	X	X	X	X	2	0	6
2	V	V	V	X	X	X	X	X	3	1	5
3	V	V	V	X	X	X	X	X	3	1	5
4	V	V	V	V	X	X	X	X	4	2	4
5	V	V	V	V	V	X	X	X	5	3	3
6	V	V	V	V	V	V	V	X	7	5	1
7	V	V	V	V	V	V	V	V	10>8	8	0
8	V	V	V	V	V	V	V	V	0	0	0

The reader can check that the *curr_hop* increases gradually in every step until it exceeds the *max_hop*. When *curr_hop* exceeds the *max_hop*, the root node sets *curr_hop* to zero and gets *fib_num* back to the first number of the Fibonacci sequence. The root node also adjusts *max_hop* to be the maximum value of hop counts of the messages that are received by the root since *max_hop* was adjusted last time.

3.2.2 An example of FiboCast

Table 2 shows an example of FiboCast. In this example, the *min_hop* is 2, the *max_hop* is 8, the first two Fibonacci numbers, *f1* and *f2*, are 0 and 1. In step 1, the message will be forwarded to the range of two hops because the *curr_hop* is 2, which is the sum of the *min_hop* (2) pluses the *fib_num* (0). In step 2, the message will be forwarded to the range of three hops because the *curr_hop* is 3, which is the sum of the *min_hop* (2) pluses the *fib_num* (1). As the Fibonacci number increases, the range that message can be forwarded increases, too. When the *curr_hop* (10) exceeds the *max_hop* (8) after step 7, the *curr_hop* sets to 0 and the first two Fibonacci numbers reset to initial value (i.e., *f1*=0 and *f2*=1). In step 8, the message can be forwarded unlimitedly until to the AOI border because the *curr_hop* is 0. After this step, the *curr_hop* resets to 2 as step 1 and repeats the previous steps. We saved 24 messages which are the sum of the Reduced fields in this example. It shows that FiboCast indeed reduces the bandwidth consumption further than VoroCast. And the neighbors near the AOI center get messages more frequently, while the farther neighbors get messages less frequently.

4 Evaluation

Our goal is to improve the AOI scalability, so we reduce the bandwidth consumption by proposed VoroCast and FiboCast to let the AOI accommodate more nodes. We use simulation to measure the performance of our schemes. And we apply several metrics, such as bandwidth consumption, neighborship consistency and drift distance, to measure the performance.

- Bandwidth consumption: It is the major metric to measure how many neighbors can be accommodated within AOI. Because of the bandwidth restriction, the node only can handle a fixed number of neighbors within its AOI. However, the fixed number of neighbors can be determined by different handling methods. The node is able to accommodate many neighbors within AOI by using a good method. In turn, using a bad method merely can accommodate few neighbors within AOI.

- Neighborship consistency: Only the AOI can accommodate many nodes is not enough. We need the consistency about the neighbor's actual position and virtual position. If the node has complete knowledge about its neighbors, the neighborship consistency is high. On the contrary, if the node only has partial knowledge about the neighbors, the neighborship consistency is low. The Figure 11 shows an example of inconsistent neighborship. The node A to J are the real neighbors of root node R and R should have the knowledge of them. But in the recognition of R, it regards the nodes that A to D, F to G, I and K' as its neighbors. So the neighborship consistency of R is low.

- Drift distance: The drift distance is another metric relative to neighborship consistency. It is the distance between the real position and virtual position of a node. The drift distance reflects the correctness of neighbor position. If the drift distance with a neighbor node is high, it indicates the root node has not the up to date information about the neighbor node. So the transmission latency is usually

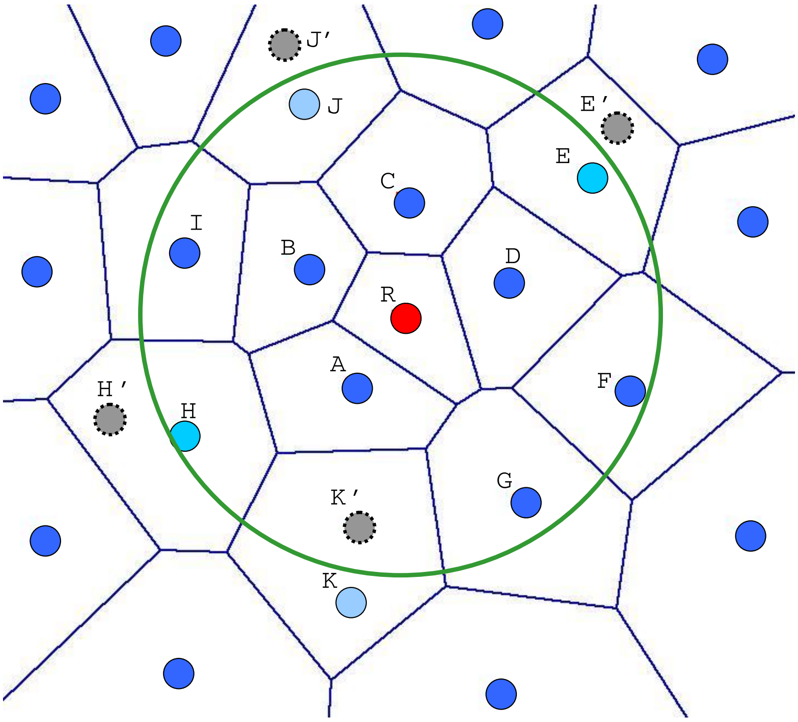

Figure 11: Inconsistent neighborship. The big circle is the AOI of root node R. The neighbor node with dotted circle is the virtual position that R knows. The node with solid circle is the actual position of the node.

the major factor affect the drift distance.

4.1 Simulation environment

We implement the VoroCast and FiboCast on top of VAST [1], which is an implementation of VON. We demonstrate the performance with generating a number of nodes and every node moves randomly on a 1000x1000 2D plane in discrete time-steps. Each node has a fixed AOI radius of 200 and generates a position update message each step. In our simulation scenario, one simulated second is ten steps and the simulation is run for 1000 steps (i.e. 100 simulated seconds). Each node has a velocity which is 5 and moves with random waypoint pattern. We assume that latency is constant and without packet loss for simplicity. The constant latency represents the sent message will be received in the next time-step and be processed. The advantage of our multicast algorithm is that it will form a spanning tree like topology, so the received messages of the root node can be aggregates at the intermediate nodes of the tree. In addition, the messages of a node will be relayed by its enclosing neighbors each step, so the messages which are relayed to the same enclosing neighbor can be packed in a packet and the data of the packet will be compressed (using zlib [2]). The aggregation and compression is benefit to the forwarding method but without benefit to directly sending method. Figure 12 shows the message compression comparison of direct sending scheme (VON) and forwarding scheme (VoroCast). directly sending scheme is not suitable to use data compression, the compression rate usually more than 90%. On the other hand, forwarding scheme is suitable for data compression. We run the following simulations from 100 to 1000 nodes in increments of 100 nodes.

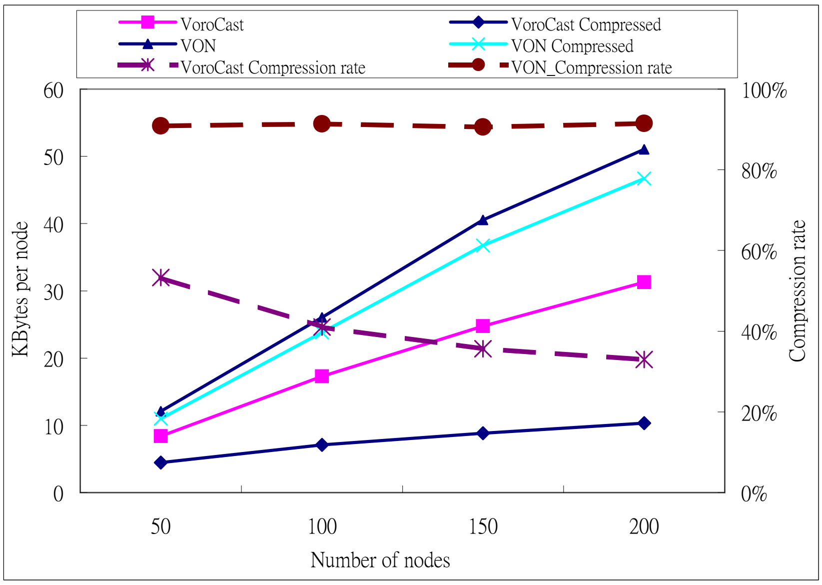

Figure 12: The compression comparison of directly sending scheme and forwarding scheme.

4.2 The simulation result

4.2.1 Bandwidth consumption

Figure 13 shows the bandwidth consumption of VoroCast and FiboCast. We can see that our schemes have less bandwidth consumption than directed-based scheme. Because the message dissemination path of VoroCast is a spanning tree, the message transmission is non-redundancy. VoroCast also apply the aggregation and data com-

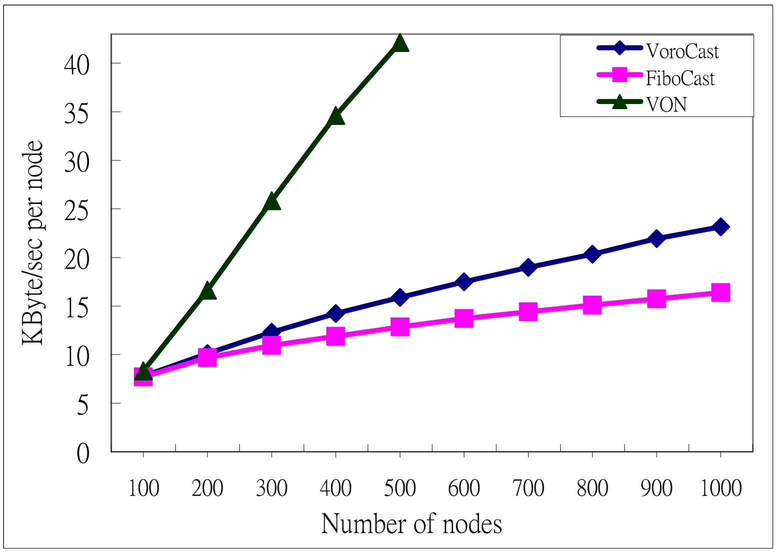

Figure 13: The average transmission size per node.

pression to reduce the transmission size. Aggregation allows several message packets share a common packet header, and data compression allows the message data size become smaller than original size. It is noted that compressing multiple messages sparsely is worse than compressing an aggregated message composed of the messages. FiboCast is based on the VoroCast, but it adjusts the message forwarding range, so it consumes less bandwidth than VoroCast.

4.2.2 The neighborship consistency

Figure 14 shows our schemes all keep a constant number of connections though the number of AOI neighbor is increasing. Our schemes balance the bandwidth because each node only has to send message to its enclosing neighbor, and the enclosing neighbors send the message to their enclosing neighbor recursively. So it does not send to all AOI neighbors at the same step. On the contrary, the number of connections is increasing with the number of AOI neighbor in directed-based scheme.

Figure 15 shows the neighborship consistency of directly sending scheme, VON, and our schemes. The message to each node in directly sending scheme is connected directly, so it can keep very high neighborship consistency. However, the message in our schemes is forwarded node by node, so the neighborship consistency is worse than

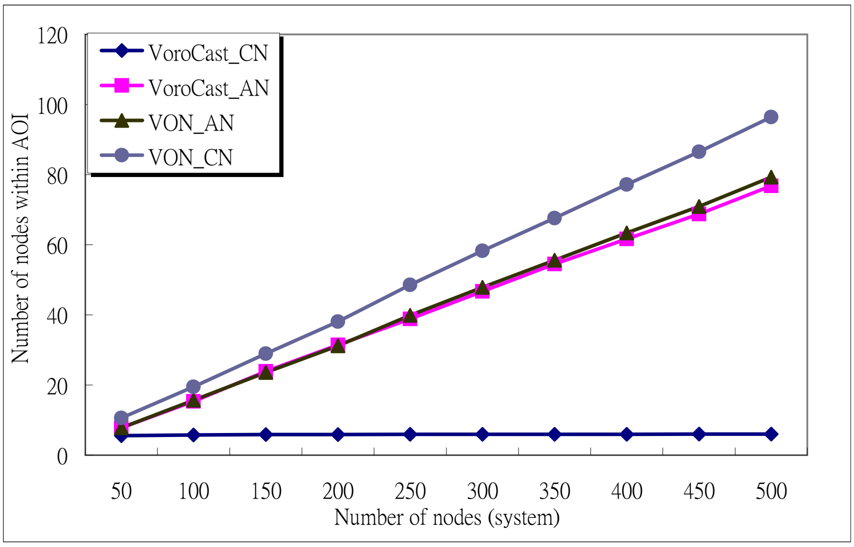

Figure 14: The comparison between AN and CN. AN is the number of AOI neighbor. CN is the number of connected neighbor.

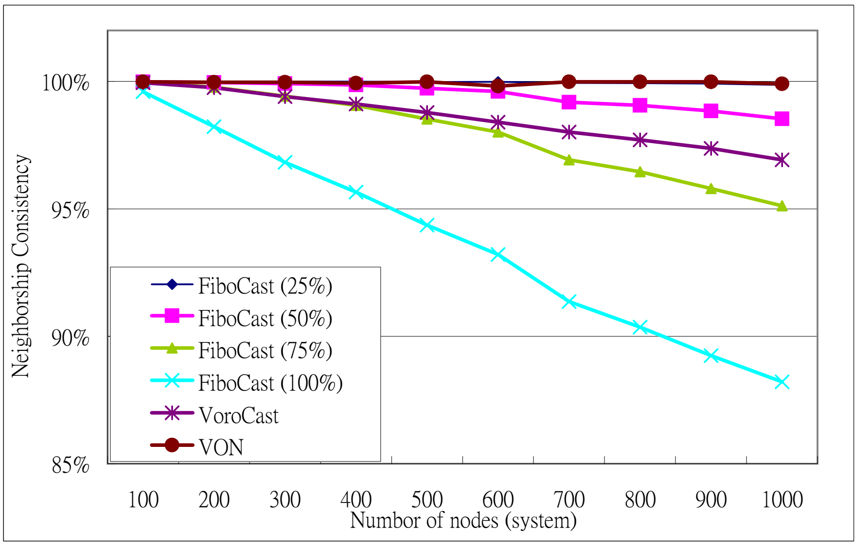

Figure 15: The neighborship consistency comparison of directly sending scheme, VoroCast and FiboCast with 4 levels. The percentage in parentheses stands for the percentage of AOI radius.

directly sending scheme. Although our schemes have worse neighbor consistency than directly sending scheme, it still has a considerable high consistency. For instance, the VoroCast has the neighborship consistency more than 95%.

The overall neighbor consistency of FiboCast is worse than VoroCast, because the FiboCast is designed for reducing the received message of the farther nodes. If we partitioned the AOI into four levels, the neighborship consistency still very good at the surrounding of center node. As Figure 15 shows, the neighborship consistency of FiboCast is still very high (exceeds 95%) within the range of 75% AOI radius. It also proves the fact that FiboCast successfully let the nearby neighbors get message more frequently, but the farther nodes get message less frequently.

4.2.3 The drift distance

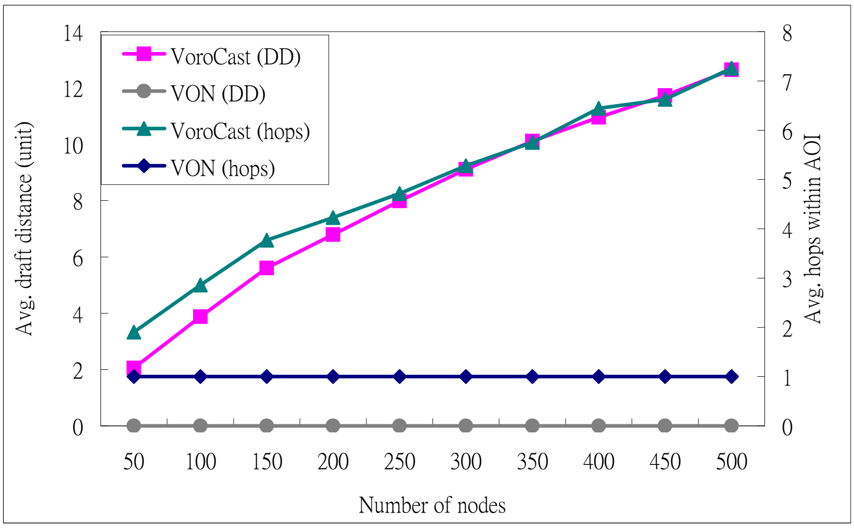

Figure 16: The relationship between drift distance and hops. DD stands for the drift distance.

We know that the drift distance reflects the difference between the reality and knowledge, of curse we would like the drift distance as small as possible. The drift distance is affected by latency seriously. However, VoroCast transmits messages by relaying, this method inherent makes received messages of each node is later than directed connection. When the number of nodes within AOI is increasing, the message has to be forwarded more hops and the drift distance is bigger and bigger (see Figure 16). From the simulation result, the drift distance of VoroCast is less than 14 even the system achieves to 500 nodes (i.e., the average hops of the message pass through is 7). It means the difference between actual position and visible position is less than 3 steps (each node moves 5 units per step). Although the drift distance on VoroCast is bigger than directly sending scheme, VoroCast can accommodate more nodes than directly sending scheme. When node is in a crowded environment, to accommodate more nodes may be more important than the little inaccuracy.

5 Conclusion

In this thesis, we proposed two schemes, VoroCast and FiboCast, to improve the AOI scalability. Among the schemes, VoroCast is based on the Voronoi diagram and intends to reduce bandwidth consumption. VoroCast reduces the bandwidth consumption not only by the non-redundant message transmission, but also by applying the aggregation and data compression mechanisms. In addition, each node selects its children by local information and forwards the messages to them.

FiboCast is useful for the AOI-cast of essential messages that have to be sent periodically in NVEs, such as position update, etc. Because users are more concerned about the surrounding events than the farther events, FiboCast dynamically adjusts the message dissemination range according to Fibonacci sequence so that the neighbors nearer the AOI center get messages more frequently, while farther neighbors get messages less frequently.

According to the simulation results, the VoroCast and FiboCast are both consume less bandwidth than directly sending scheme. Besides, FiboCast consumes less bandwidth than VoroCast, the overall neighborship consistency of FiboCast is worse than the neighborship consistency of VoroCast. But if we analysis it at length, the neighborship consistency near the source node (within the area of 75% AOI) still has high neighborship consistency (i.e., exceeds 95%).

Although our schemes are able to consume less bandwidth, the transmission latency does not reach the optimal case. In some worst case, it may be very high. We will continue to investigate how to lower the transmission latency to optimization. Then it will make our scheme more efficient.

6 Reference

References

[1] VAST Project. *http://vast.sourceforge.net/*.

[2] zlib Project. *http://www.zlib.net/*.

[3] Franz Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.

[4] M. Castro, MB Jones, A.M. Kermarrec, A. Rowstron, M. Theimer, H. Wang, and A. Wolman. An evaluation of scalable application-level multicast built using peer-to-peer overlays. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, 2.

[5] Tsu-Han Chen. A Forwarding Model for Voronoi-based Overlay Network. Master's thesis, Tamkang University, Taiwan, 2006.

[6] S.Y. Hu, J.F. Chen, and T.H. Chen. VON: a scalable peer-to-peer network for virtual environments. *Network, IEEE*, 20(4):22–31, 2006.

[7] Sunghwan Ihm Tcaesvk Gim Jinwon Lee, Hyonik Lee and Junehwa Song. APOLO: Ad-hoc Peer-to-Peer Overlay Network for Massively Multi-player Online Games. Technical report, Korea Advanced Institute of Science and Technology (KAIST), 2006.

[8] Y. Kawahara, T. Aoyama, and H. Morikawa. A Peer-to-Peer Message Exchange Scheme for Large-Scale Networked Virtual Environments. *Telecommunication Systems*, 25(3):353–370, 2004.

[9] J. Keller and G. Simon. Toward a peer-to-peer shared virtual reality. *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, pages 695–700, 2002.

[10] J. Keller and G. Simon. Solipsis: A massively multi-participant virtual world. *Proc. of PDPTA*, pages 262–268, 2003.

[11] B. Knutsson, H. Lu, W. Xu, and B. Hopkins. Peer-to-peer support for massively multiplayer games. *IEEE Infocom*, 2004.

[12] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, 11:329–350, 2001.