A Forwarding Model for Voronoi-based Overlay Network

Jui-Fa Chen, *Wei-Chuan Lin, Tsu-Han Chen, Shun-Yun Hu
Department of Computer Science and Information Engineering,
TamKang University, Danshui, Taiwan 25137, R.O.C
E-mail: alpha@mail.tku.edu.tw
*Department of Information Technology,
*Takming University of Science and Technology,
*E-mail: wayne@takming.edu.tw

Abstract

An approach to build highly scalable and robust networked virtual environments (NVEs) is using the peer-to-peer overlay networks. Voronoi-based Overlay Network (VON) has been proposed to maintain a highly overlay topology consistency in a bandwidth-efficient manner. However, the related research requires all nodes to connect directly with their relevant neighbors. This limits the number of neighbors should appear within the area of interest (AOI) of a given node. A new forwarding model for VON is proposed to solve this problem by connecting only with the nearest neighbors that is called the enclosing neighbors (EN), and propagating the position updates information to other nodes by message forwarding. In this way, the AOI of a given node may be more flexibly expanded and different bandwidth capacities may be more efficiently utilized.

Keywords: Peer-to-peer (P2P), overlay network, networked virtual environment (NVE), Voronoi-based Overlay Network (VON)

1 Introduction

Massively Multiplayer Online Game (MMOG) [6] is a class of networked virtual environments (NVEs). It is rapid growth in recent years and most of them are using the client-server architecture. However, scalability of such systems face challenges such as the server design complexity, hardware-provisioning, load-balancing and fault-tolerance issues. Efforts have recently been put by utilizing peer-to-peer (P2P) networks to support such large-scale NVE systems [1, 2, 3, 4]. Among these proposals, Voronoi-based Overlay Network (VON) promises to provide scalability and overlay topology consistency in a bandwidth-efficient manner [4, 5]. However, the current proposal requires each node to connect directly to all neighbor nodes of interest which called Area-of-Interest, or AOI neighbors (ANs). This approach is called the direct-connection model and required two nodes to have mutually visibility. However, in a realistic network environment, every node has different network bandwidth, and it is better that if each node can independently adjust its visible area and transmission size according to its capacity. This paper proposes a new design to reduce the number of connections and compress the collected message to reduce transmission size. The propose mechanism is called the forwarding model, that is an extension to the original direct-connection model of VON. Forwarding model is allowing messages relayed by neighbor nodes, and restricting each node's direct connections with only its nearest neighbors called the enclosing neighbors (ENs). By using message forwarding and data compression, each node may potentially see a larger number of ANs and more efficiently utilize its bandwidth resource.

The rest of this paper is explained as follows. Section 2 is the related work about P2P based NVE. Section 3 is the proposed algorithm of the forwarding model. Section 4 is the simulation results. Section 5 is the discussions of the simulation results. Section 6 is the conclusion.

2 Related Works

Peer-to-peer network is a non client-sever network topology. Data is transferred without centralized severs in a distributed fashion. In other words, every machine is not only a client, but also a server. An important issue is that to whom each node should connect to and communicate with in a P2P network. Voronoi diagram is a mathematical construct such that given n nodes on a 2-D plane; these n nodes can partition the plane into n non-overlapping Voronoi regions [4]. The enclosing neighbors (ENs) are defined as those nodes whose regions immediately surround a particular node. In Figure 1, if node S is the node in discussion, then all the pink nodes are its

EN, because their Voronoi regions immediately surround S's Voronoi region. When nodes moved, each node's ENs will update frequently. The AOI (area of interest) of S is also defined as the area indicated by the green circle. Nodes within the AOI are called AOI neighbors (ANs), which are the nodes S got interested to. In Figure 1, the pink nodes and green nodes are both ANs of S. Boundary neighbors (BNs), which are shown as the red nodes, are node S's ANs which are only partially inside S's AOI.

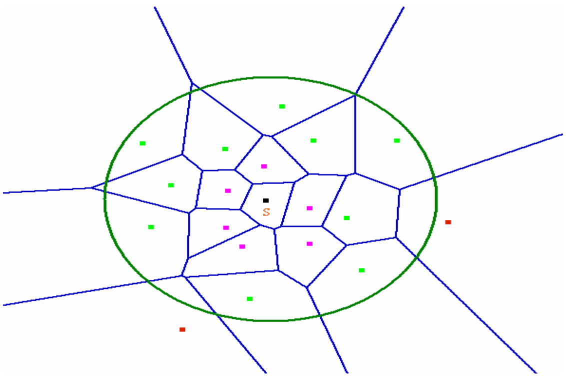

Figure 1: Voronoi diagram

VON is a P2P NVE building by Voronoi diagram. It supports nodes to join and leave dynamically; nodes can move in this NVE, and interact with their ANs. Keeping the ENs correctly is a mainly promise to use the architecture. The direct-connection model is an established model by using this architecture [4]. In this model, nodes make direct connection with their ANs, and send position updates to their ANs at each time-step. As nodes move, new neighbors are discovered by notification from the boundary neighbors, as they know both the moving node and the other potentially visible new ANs. Details of the direct-connection model can be found in [4] and [5]. However, as nodes need to connect to all of their ANs directly, if there are too many ANs, the transmission size will be too big to transfer in time. Direct-connection model also requires all nodes to be mutually visible, which decreases visibility unnecessarily for nodes that have larger bandwidth capacity.

Forwarding model is proposed to improve the problems described in direct-connection model. The proposed model possesses the following characteristics:

a. Forwarding model is a kind of parallel transmission mechanism. This mechanism would not have the bottleneck in limiting the transmission number.

b. Forwarding model is to provide the relation of the peer's AOI and the transmission ability. If the transmission ability of a peer is weaker, its AOI radius should be adjusted to improve the transmission ability.

c. Through the adjustment of peer's AOI radius, the quantity of transmission is reduced and can let peers with different abilities to interact in this proposed virtual environment.

3 FORWARDING MODEL ALGORITHM

The main idea of the forwarding model is that if each node only directly connects to its ENs, who should forward relevant messages to other nodes? This would achieve larger AOI visibility with less number of active connections. If sending message by forwarding method, the massages can be compressed before sending to the node's ENs to reduce the network bandwidth. To make sure that messages can be forwarded by ENs, three major procedures would be described. The first is make sure that all nodes know their ENs correctly. Second, this environment needs to update the enclosing neighbor list and gets other ANs' state messages when nodes are moving around in this VON (MOVE procedure). Finally, how to exit this P2P overlay environment is also important (LEAVE procedure).

3.1 Join

3.1.1 Forwarding "JOIN" message

A node joins the VON (called joiner) by sending a "JOIN" message, which contains initial position, AOI radius and its IP/port information to a gateway. This gateway could be the creator of this overlay network or some existing nodes in the VON. The message is forwarded until reaching an acceptor, which is the node whose Voronoi region contains the joiner's initial position as shown in Figure 2.

3.1.2 Initial Neighbor List

If the joiner is acceptor's EN, the acceptor will send what it considers as the joiner's enclosing neighbor list to the joiner as shown in Figure 2. As the joiner's enclosing neighbor list may not be complete, the joiner will send a "HELLO" message to its initial ENs to make sure that its enclosing neighbor list is correct. The "HELLO" message contains the joiner's states such as its position, AOI radius and IP address and a list of the ENs as it knows. When the new neighbors find that the list is incomplete, they would send the missing ENs' states to the joiner. The same procedure would repeat with each new neighbor discovered, until no more missing ENs are found.

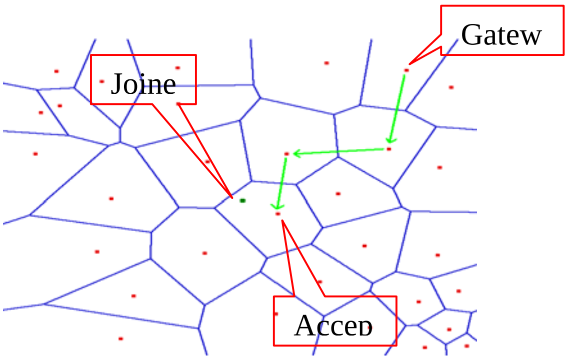

Figure 2: JOIN message diagram

3.2 Move

3.2.1 Maintaining Enclosing Neighbors

When nodes move, their neighbors would change, so the environment needs to record the ENs' overlap neighbors to discover the nodes' ENs. The overlap neighbor of a moving node S's is the neighbor shared for both S and A as shown in blue region of Figure 3(a). When the overlap neighbors change, the moving nodes should be notified about these changes. The moving node would determine whether these newly discovered neighbors are really its new ENs or not. In Figure 3, 3(a) shows time-step 1 and 3(b) shows time-step 2. At time-step 1, the overlap neighbors of S and A are B and C. At time-step 2, as D has moved in, so the overlap neighbors of S and A change to B and D. At this time, A would notify S about D's information.

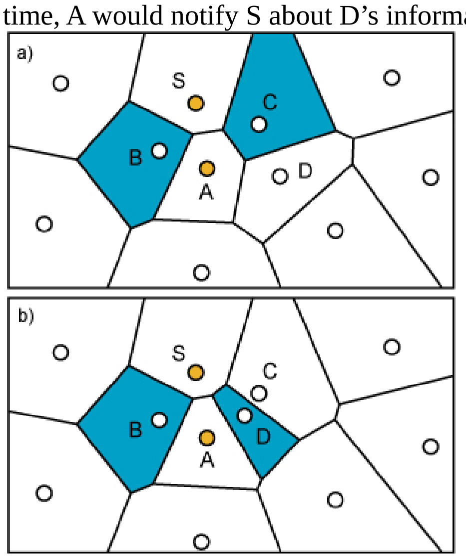

Figure 3(a) and (b): Movement of Node D

3.2.2 Maintaining ANs

There are two actions to maintain the node's ANs, one is "publish" and the other is "subscribe". The action of "publish" is to expand the node's message to all of subscribers. The "publisher" is the original node to execute the action of "publish". The action of "subscribe" is the node to tell its ANs about the needed information. The "subscriber" is the original node to execute the action of "subscribe". The actions of subscribe and publish are done simultaneously. The action of "subscribe" is to sent the message to its ANs and become ANs' subscriber. Whiles the action of "publish" is to send the message to all of the subscribers. Therefore, the node would send its information to its ANs and subscribers in every time step. In the meantime, this node also registers to its ANs and sends message to all of the registers.

Every node should know all of the neighbors in its AOI so as to forward nodes' message to its ANs. That is these nodes would send their message to their subscribers. These steps are shown as Figure 4. The circle in dashed line is the AOI of the node in red of Figure 4(a) and (b).

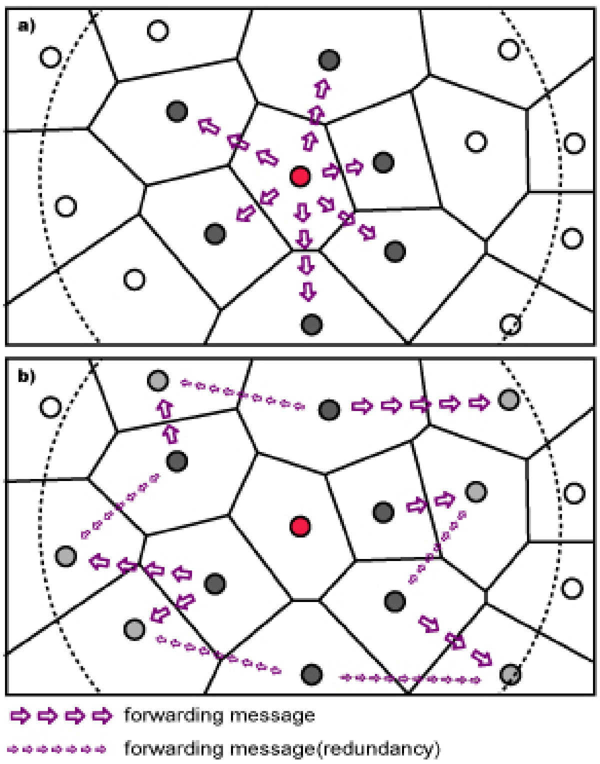

Figure 4(a) and (b): Forwarding Message

The detail procedures for maintaining a node's AN is explained as followed:

First, a moving node would broadcast its position to its ENs. “Broadcast” does not mean that the hardware-level broadcast, but sending messages separately to the entire node’s ENs in one time-step. Then its ENs would decide whether the message should be forwarded by checking if there are still ANs that have not received the message as shown in Figure 4(b). If the message needs to be forwarded, the node would forward the message to its own ENs and create a forward path during the repetition of this process. The forward path is a spanning tree and the message’s publisher is the root of this tree. A forward path consists of a series of forward records as shown in Figure 5 which are kept at each of the intermediate forwarder.

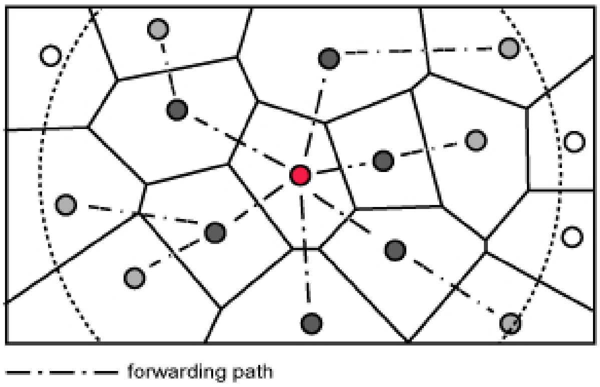

Figure 5: Forward Records

Each record is composed of a pair of destination’s id and previous forwarder’s id in a (destination, forward to) dataset as shown in Figure 6. “Destination” indicates the moving node or, equivalently, the update message’s publisher and “forward to” indicates the previous node which forwards this message. When receiving messages, the publisher of an update message is a subscriber of this message’s receivers. This process is repeated until the message no longer needs to be forwarded. This action also notifies all the ANs of the moving node about its current position.

Second, if a position update message is received, the receiving node would check whether the sender is a valid AN or not and process accordingly. There are two scenarios which need to discuss as shown below:

A. Two nodes are not mutually visible.

When an node broadcasts its position, it should add other nodes’ ids which is interested in this information (i.e. the nodes whose AOIs cover the broadcasting node). When the message is sent to other forwarders, all of the interested nodes may properly receive this position update message even if they are outside of the broadcasting node’s AOI. For example, as shown in Figure 6, after S broadcasts its message, a forward path of (S, A, B, P) is built by nodes A, B and P in a distributed fashion. A’s forwarding record is (S, S), B’s forwarding record is (S, A), and P’s forwarding record is (S, B). Assuming that A is not P’s AN, and P is S’s AN, when it is node P’s turn to send the position update message, it would include the id of S as one of the targets of its update information nodes. Therefore, node P’s message will be forwarded to S. When B receives P’s message, it would search from its forwarding records to find if the message needs to be forwarded. Then B would forward the information to A because of the forwarding record (S, A). This process continues along with the forwarding path and ensures that S eventually receives P’s position update.

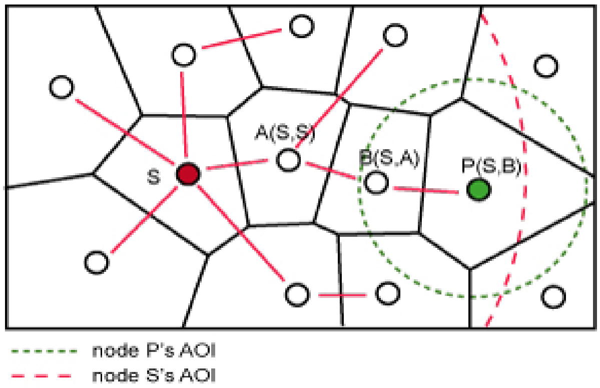

Figure 6: Example for two nodes that are not mutually visible

B. Two nodes are mutually visible.

Nodes will simply broadcast their messages. As message update would be propagated to the boundary of AOI, mutually visible nodes would receive each other’s position update message without problems. For example, assuming A and B are both within each other’s AOI, then when A broadcasts its message and B will receive it, and vice versa.

3.2.3 Forwarding Optimize

Forwarding model has redundant transfer problem when forwarding messages as shown in Figure 4(b). This problem would increase the message transmission times. The optimize mechanism is mainly to improve this phenomenon. There are two methods proposed to reduce this transmission redundancy as discussed below:

A. Forwarding by nodes whose ID numbers are closer to its EN

It is a simple method to reduce transmission redundancy by checking the ENs' ID number. Consider that node S and another node P both send the same message to node F, if node F's ID number is closer to node S than P, then S would be allowed to send this message.

B. Forwarding by nodes whose AOI are bigger

It works when nodes having different AOI-radius. The node would find a suitable forwarder in its enclosing neighbor list. In Figure 3(a), node B and C receive S's message form A, B knows C also receive S's message because C is A's EN. In Figure 3(b), when B tries to forward S's message to D, B would detect that node C is more suitable to forward the message. Which node is more suitable to forward would be decided by their AOI, the node with bigger AOI is more suitable to forward message. The nodes with smaller AOI would be skipped by comparing the AOI with another forwarding node. By adjusting AOI-radius, the proposed mechanism can control the transmission times.

3.3 Leave

For leaving this VON, a node simply disconnects. The leaving node's neighbors would recover the proper topology and discover new ANs, by following the regular MOVE procedure. When a node is left, the Voronoi diagram of the virtual environment should be redrawn to find the new ENs and sent "HELLO" message to these new ENs. As shown in Figure 7 (a), node A is left and the the Voronoi diagram of the virtual environment should be redrawn as shown in Figure 7 (b) to maintain the node consistency.

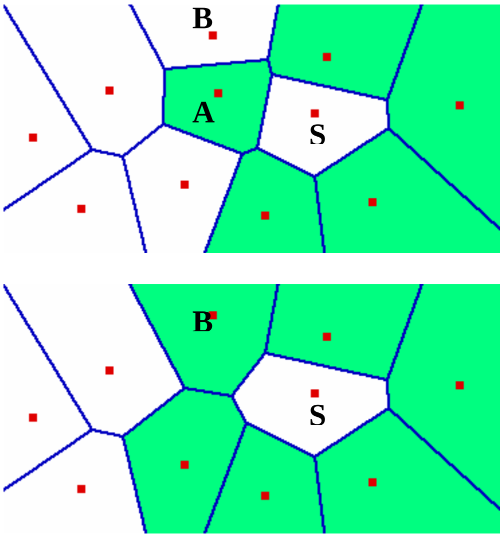

Figure 7(a) and (b): Redraw Voronoi diagram after Node Leave

4 SIMULATION

4.1 Overview

This simulation shows the characteristics of forwarding model and compares the node consistency, drift distance, transmission size and connect neighbors with the direct-connection model. Node consistency can show whether the node could connect the correct ANs or not. Drift distance records the average drift distance of nodes' ANs. If the drift distance is growing up, it means that there are updating messages that are transmitted delay.

4.2 Simulation Methods and Metrics

A forwarding model is build on VAST [7]. All of the nodes random move in a 2-dimension map and the messages update 10 times per second. The transmission mechanism in forwarding model uses the data compression with zlib [8]. Whether to use data compression or not is decided by the size of message. As the message in direct-connection model is sent separately, the message size would be bigger as compared with the forwarding model. However, in forwarding model, because the message is collected before sending, it is more suitable to use the data compression. The settings of the simulation runtime environment are shown as Table 1 and 2.

Table 1. The setting in Figure 9-12

World dimension	1000 x 1000
Simulation steps	1000
AOI-radius	200
Simulation size	50 to 500, with an increment of 50

Table 2. The setting in Figure 13

World dimension	1000 x 1000
Simulation steps	1000
AOI-radius	10 to 200, with an increment of 10
Simulation size	200

4.3 Simulation Results

4.3.1 Node Consistency

Node consistency is used to measure that whether each node can receive the message from the ANs or not. The definition of node consistency in this paper is defined by Kawahara's paper [3].

$$\text{Consistency}(\%) = \frac{1}{N}\sum_{i=1}^{N}\frac{P(i)}{Q(i)} \quad ,\ldots\text{(Formula 4.1)}$$

In Formula 4.1, N is the number of node in the virtual environment, P(i) is the actual number of ANs received by node i, Q(i) is the total number of ANs that should be received by node node i. Through the computing of the node consistency, the nodes in the virtual environment can check whether all of the ANs can be known correctly or not. Node consistency in forwarding model is lower than direct-connection model because the forwarding model needs more times to message forwarding. The result is shown in Figure 8.

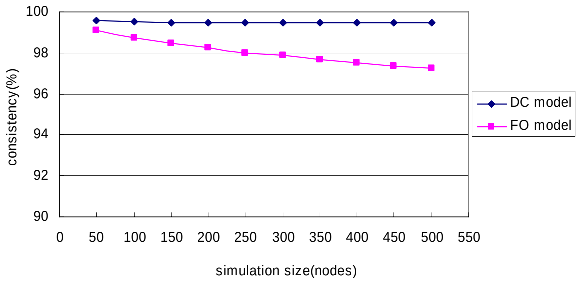

Figure 8: Comparison of the node consistency

4.3.2 Drift distance

The computation of the drift distance of node is to show the correct rate of the received node's position. If the value of the drift distance is high, the result is shown that the received information of node is not suitable to be referenced [6]. The computation of the drift distance is shown as Formula 4.2. In formula 4.2, drift_dist(i) is to compute the average drift distance of AN of node i. Drift distance in forwarding model is also higher than direct-connection model, it is caused by the increasing of forwarding times. Figure 9 shows the comparison results of forwarding and direct-connection model. Figure 10 is shown the relation of drift distance and the message forwarding times.

$$\text{Drift distance} = \frac{1}{N}\sum_{i=0}^{N} drift\_dist(i) \text{, (Formula 4.2)}$$

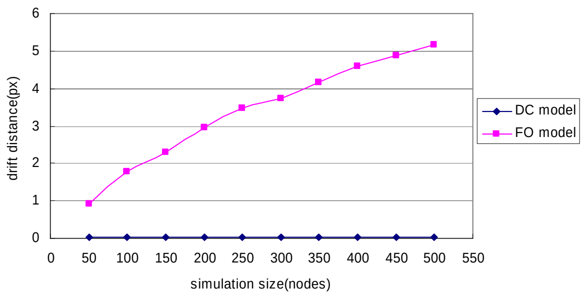

Figure 9: Comparison of the drift distance

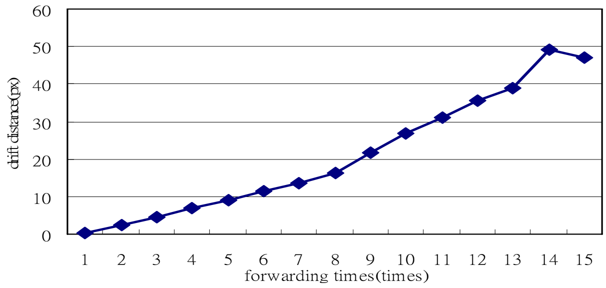

Figure 10: Relation of drift distance and forwarding times

4.3.3 Transmission size

By using the message compression, transmission size in forwarding model is lower than direct-connection model as shown in Figure 11.

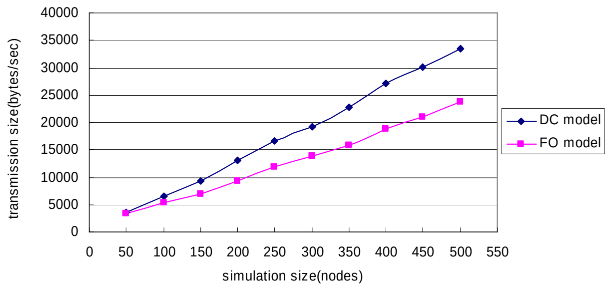

Figure 11: Relation of transmission size and simulation size

The forwarding optimize method B in 3.2.3 is simulated to prove the positive relation between transmission size and AOI-radius after using the forwarding optimize method as shown in Figure 12.

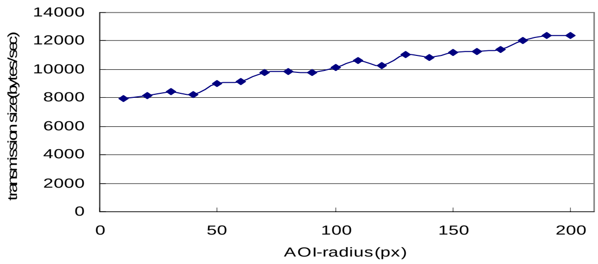

Figure 12: Transmission size after forwarding optimize method B

5 DISCUSSION

- **Overlay partition in VON**

Overlay partition refers to VON is cut into two or more separate partitions, and nodes in these partitions do not know each other. The key point of keeping VON alive is to ensure the full connectivity of all nodes and to maintain the Voronoi diagram correctly.

Every scenario that makes Voronoi diagram incorrect has the possibility to cause overlay partition in VON. However, it is possible for overlay partition to occur. For example, if nodes move too fast and cause a given node's ENs to change completely in one time step, then neighbor discovery may not be completed and cause the phenomenon of overlay partition.

- **VON self-recover from inconsistency**

When inconsistency occurs by packet lost or other causes, if the ENs list is kept correctly, VON still can be recovered by the neighbor list. However, if there are too many nodes need to recover their neighbor list, VON overlay partition may happen. Figure 13 shows the consistency variation after 5000 steps. The consistency is stable at 98%, and this proves that the inconsistency can be recovered by VON.

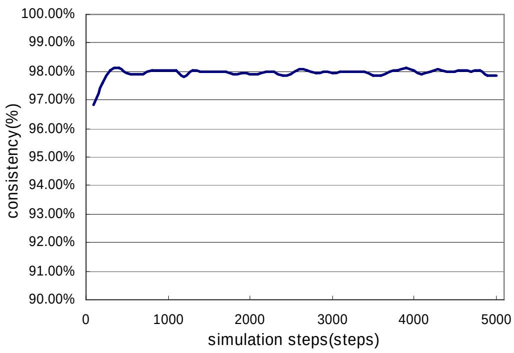

Figure 13: Consistency stable

Finally, the advantage of forwarding model can be shown as followed:

1. Low connection
   As compared with the direct-connection model, the node of this proposed mechanism just needs to connect with its ENs. This would make the connection of nodes simple.
2. Data collection
   Because just connects with its ENs, each node could gather all the information and then to transmit, this would also utilize the compress method to reduce the size of transmission information.
3. Parallel transmission
   Due to the transmission of information is through ENs, the proposed mechanism also uses the parallel transmission to improve the bottleneck of hardware transmission.

6 CONCLUSION

In this paper, the proposed forwarding model of VON is explained and the simulation results are shown. By using data compression mechanism to send the message, the proposed forwarding model behaves better than direct-connection model in transmission size and message transmission times. However, in the drift distance and node consistency cases, the proposed forwarding model behaves not as better as direct-connection model. The proposed forwarding model is suitable to the virtual environment that a node with large number of ANs and low network latency.

In the future, the forwarding model can be improved in the drift distance and node consistency cases. Combining the direct-connection and the proposed forwarding models is another goal to achieve that VON may be adapted to larger node number of network environments.

REFERERNCE

[1] J. Keller and G. Simon, "Solipsis: A massively multi-participant virtual world," in Proc. Int. Conf. Parallel and Distributed Techniques and Applications (PDPTA 03), pp. 262-268, 2003.
[2] B. Knutsson et al., "Peer-to-peer support for massively multiplayer games," in Proc. IEEE INFOCOM, pp. 96-107, Mar. 2004.
[3] Y. Kawahara, T. Aoyama, and H. Morikawa, "A peer-to-peer message exchange scheme for large-scale networked virtual environments", Telecommunication Systems, vol. 25, no. 3-4, pp. 353–370, 2004.
[4] S. Y. Hu and Jui-Fa Chen, "VON: A scalable peer-to-peer network for virtual environments", IEEE Network, vol. 25, no.4, pp. 22-31, 2006.
[5] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, "Spatial tessellations : concepts and applications of Voronoi diagrams," Chichester, Wiley. 2nd ed., 2000
[6] C. Diot and L. Gautier, "A distributed architecture for multiplayer interactive applications on the Internet", IEEE Network, vol. 13, no. 4, pp. 6-15, 1999
[7] VAST Project homepage. Available at: http://vast.sourceforge.net/
[8] Zlib Project homepage. Available at: http://www.zlib.net/