Enhancing Neighborship Consistency for Peer-to-Peer Distributed Virtual Environments

Jehn-Ruey Jiang, Jiun-Shiang Chiou and Shun-Yun Hu

Department of Computer Science and Information Engineering,
National Central University
Jhongli City, Taoyuan, 320, Taiwan

Abstract

Although P2P-based DVEs can potentially solve the scalability issues, other important issues such as consistency remain unexplored. In this paper, we address neighborship consistency, which is the ratio between the number of known nodes and the number of actual nodes within a node's area of interest (AOI). We address two factors that affect neighborship consistency for DVEs using Voronoi-based Overlay Network (VON). We propose adaptive AOI buffer and critical node detection to reduce the negative impacts of the two factors. We also perform simulation experiments to demonstrate the effects of the two factors before and after adopting the proposed mechanisms.

1 Introduction

Distributed Virtual Environments (DVEs) are computer-generated virtual world where multiple geographically distributed users can assume virtual representatives (or avatars) to concurrently interact with each other [SZ99] [SKH01]. Examples of DVEs include early DARPA SIMNET and DIS systems [MT95] as well as currently booming Massively Multiplayer Online Games (MMOGs). For some DVEs with a great deal of concurrent users, such as MMOGs, traditional client/server (C/S) architecture may have the problem of scalability due to the limitation of server resources. To increase the scalability, several recent approaches [HCC06], [KS03], [KAM04], [MKMA04] utilize peer-to-peer (P2P) architecture as the foundation of DVEs. Such P2P DVEs can potentially achieve high scalability with an extremely large number of users (e.g. over one million) by distributing the server load to all participants.

One fundamental problem of DVEs is to keep consistency, such as user-state consistency, event-ordering consistency, etc. In this paper, we focus on neighborship consistency for P2P DVEs. Each node (user or avatar) in a DVE has and area of interest (AOI), which is usually a disk-shape area centered on the node. All the nodes in a node's AOI is said to be its *neighbors*. A node should be aware of the existence and variations of all its neighbors. Therefore, we should keep as high as possible the neighborship consistency, which is defined to be the ratio between known neighbors and actual neighbors. Low neighborship consistency may make a node miss some events within its AOI, which in turn may make the DVE work abnormally.

In traditional DVEs of the client/server (C/S) architecture, keep high neighborship consistency is trivial because all user states in the system are maintained by a centralized server (or server cluster). In P2P DVEs, all states are maintained by participating nodes. It is thus harder to keep high neighborship consistency. Papers [KAM04], [KS03], [MKMA04] and [HCC06] propose mechanisms trying to achieve high neighborship consistency in P2P DVEs. Among them, the mechanism based on Voronoi-based Overlay Network (VON) [HCC06] has the best neighborship consistency and comparably low control overhead and latency.

In this paper, we focus on enhancing the neighborship consistency of VON-based DVEs. We first address two factors which could influence neighborship consistency, namely *node mobility* and *node failure*. We then propose mechanisms to reduce the negative impact of the factors. We also perform simulation experiments to demonstrate that the proposed mechanisms improve VON-based DVEs dramatically in terms of neighborship consistency.

The rest of this paper is organized as follows. In section 2, we give some preliminaries. Section 3 illustrates the two factors that affect neighborship consistency; section 4 discusses and gives mechanisms to eliminate their impact. The simulation results are demonstrated in section 5 and section 6 concludes this paper.

2. Preliminaries

2.1 P2P-Based DVEs

Several approaches [HCC06], [KS03], [KAM04], [MKMA04] utilize P2P architecture as the foundation of DVEs to increase the scalability. Such P2P DVEs can potentially achieve high scalability to accommodate an extremely large number of users by distributing the server load to all participants. With cooperation among participants, P2P DVEs should provide quick and efficient neighbor discovery mechanisms for each participating node to timely perceive its surroundings. The system in [KS03] uses the concept of the convex hull to ensure the success of neighbor discovery. However, it is shown to have low neighborship consistency and long neighbor discovery latency. The systems in [KAM04] and [MKMA04] require a node to directly connect to some nearest neighbors (in terms of Euclidian distance in DVE space) to exchange neighbor lists for neighbor discovery. However, the network traffic is usually very high for such systems to achieve high neighborship consistency. The paper [HCC06] proposes a system called VON (Voronoi-Based Overlay Network) to use the concept of Voronoi diagrams for neighbor discovery. As shown in [HCC06], VON usually has low communication overhead, short latency and high neighborship consistency. For all the merits of VON, we therefore focus on it in the following context.

VON uses the concept of Voronoi diagram [A91] to divide the 2D DVE plane into *Voronoi regions*. Each node *i* has an associated Voronoi region, and has three types of neighbors. Node *i*'s *AOI neighbors* are the nodes within *i*'s AOI; its enclosing neighbors are nodes whose regions directly surround *i*'s region, and boundary neighbors are nodes whose regions overlap with the AOI boundary (please see Figure 1).

Every node maintains a Voronoi diagram of all its three types of neighbors and connects directly to them. VON-based DVE also adopts the concept of *dynamic AOI* to shrink or expand a node's AOI so that the number of the node's neighbors will not exceed a pre-specified threshold. To prevent a node from becoming isolated in a sparse DVE, a node has to connect to all its enclosing neighbors at least. When a node moves, it will send its position update information to all connected neighbors. A node i's boundary neighbors will perform neighbor discovery on behalf of i because they connect to i and knows some nodes outside i's AOI that are i's potential neighbors (those potential neighbors happen to be the enclosing neighbors of the boundary neighbors).

2.2 Neighborship Consistency

In light of [KAM04], we define *neighborship consistency* as follows:

$$NC_i = \frac{NN_i}{N_i} \quad (1)$$

$$NC = \frac{\sum_{i=1}^{n} NC_i}{n} \quad (2)$$

NC_i is the neighborship consistency of node i. It is defined to be the ratio of NN_i to N_i , where NN_i is the number of AOI neighbors known by node i and N_i is the number of actual AOI neighbors of node i. NC is the neighborship consistency of the system and n is the total number of nodes. For example, if a node has 100 AOI neighbors but it is only aware of 90 AOI neighbors, its neighborship consistency is 90%. Each node's neighborship consistency can be computed in this way and then the neighborship consistency of the system can be calculated by averaging all nodes' neighborship consistency.

Neighbor consistency can be used to measure the quality or connectivity of a system. The ideal neighborship consistency should be apparently 100%, but several factors will it. When neighborship consistency is not 100%, it means that some nodes do not know all of their neighbors. In this situation, a DVE system may work abnormally. Therefore, it is essential to keep neighborship consistency as high as possible.

2.3 Overlay Partition

Overlay partition is a situation that the overlay is divided into two or more groups (partitions) and nodes in different groups do not aware of one another. In such a case, two or more partitions will become isolated worlds and neighborship consistency could be damaged seriously.

Due to the dynamic nature of P2P systems, all existent P2P DVEs suffer from overlay partition. The paper [KAM04] uses a dedicated centralized server to perform *random introduction* to reduce the negative impact of overlay partition. The drawback of this method is the need of the dedicated centralized server. The paper [MKMA04] uses an additional DHT-based overlay to handle overlay partition. However, the maintenance of the DHT-based overlay incurs much overhead for the system.

In VON, overlay partition may occur when nodes gather into some specific areas. In such a situation, there may be few nodes between the areas. Take the DVE plane in Figure 2 for example, nodes gather in left and right sides of the plane and three nodes are in the central area. Note that the nodes in the left and the right sides are aware of one another through the nodes in the central area. If the three nodes in the central area fail simultaneously, overlay partition occurs; nodes in the left side and nodes in the right sides cannot be aware of one another anymore.

3 Factors Affecting Neighborship Consistency

In this section, we identify two factors that influence the neighborship consistency in DVEs, especially for DVEs based on VON.

3.1 Node Mobility

In DVEs, each node usually moves around the environment. The node mobility may have negative impact on neighborship consistency. For example, when a node j moves very fast, it will enter a node i's AOI and comes out in a short time. Because node j moves very fast, node i may not know that j has ever entered its AOI. This will bring down neighborship consistency.

When nodes with very high speed move in DVEs, it is difficult for other nodes to notify the moving node of new neighbors and vice versa, resulting in low neighborship consistency. This is because the speedy nodes could move across many nodes' AOIs in a short time. However, when a node stay within some other node's AOI for very short period, there may not be any interaction between the nodes. Hence, we may well only consider nodes with moderate mobility.

3.2 Node Failure

Node failure in P2P DVEs is a common factor because dynamically joining/leaving of nodes can be regarded as node failure. It goes without saying that node failure has negative impact on neighborship consistency. In particular, if failure leads to overlay partition, the neighborship consistency is damaged seriously. Overlay partition is very difficult to detect in P2P DVEs because the detection needs a global knowledge of all nodes' states.

Although VON is robust enough (as seen from the simulation results in [HCC06]), overlay partition may still occur when there are simultaneous failing nodes. When nodes do not spread uniformly but cluster into certain areas, the simultaneous failing nodes between those areas could lead to overlay partition. This is because nodes in different areas only can become aware of one another through the nodes between the areas.

4 Proposed Mechanisms

In this section, we propose mechanisms to reduce the negative impact caused by the two factors of node mobility and node failure for VON-based DVEs.

4.1 Conquering Node Mobility

We suggest using the mechanism of *adaptive AOI buffer* to add a donut-like buffer outside the AOI for reducing the negative impact of node mobility on neighborship consistency. To be more precise, if the speed of a node is d, then its AOI buffer will also be of width d. Note that the node's speed d is measured by the moving distance of the node during the period of two consecutive position updates. For example, if a node moves 20 units per time step and the time interval for the node to update its position is 2 time steps, then the speed of the node is regarded as 40 (units per position update). Therefore, the node should have an AOI buffer of width 40 units. Since a node is assumed to send its position update per time step, the width of AOI buffer and the speed usually have the same value. Please see Figure 3 for the illustration of the adaptive AOI buffer mechanism.

All nodes in a node i's AOI buffer will also be considered as i's AOI neighbors, which should be directed connected by i. As i's speed is getting higher, the width of the AOI buffer is getting wider, which in turn leads to more directed connected neighbors. It is noted that each node should conform to the limit of direct connected neighbors to prevent a node from using up available outgoing bandwidth.

4.2 Conquering Node Failure

The third factor is node failure. As we have shown, node failure may cause overlay partition, which will damage neighborship consistency seriously. Take the DVE in Figure 4 for example. There are 5 clusters of nodes in the DVE plane. Few nodes, those marked with AOI circles (○), are between each pair of clusters; they are the bridges to connect the nodes between clusters. If some of these nodes fail at the same time, overlay partition may then occur. These nodes marked with circles are called *critical nodes*.

In VON-based DVEs, nodes will exchange their connected neighbor lists periodically. By the neighbor lists, a node can derive useful information to decide if itself is a critical node. Below, we suggest using the *neighbor level (NL)* for identifying critical nodes. We define the neighbor level as follows:

$$NL_i = \frac{NN_i}{NN_{AVG}} \tag{3}$$

$$NN_{AVG} = \frac{NN_{SUM}}{NN_i} \tag{4}$$

NL_i is the neighbor level for node i and NN_i is the number of i's AOI neighbors. NN_{AVG} is defined to be the ratio of NN_{SUM} to NN_i, where NN_{SUM} is the summation of the numbers of AOI neighbors for all i's AOI neighbors. By equations (3) and (4), we have

$$NL_i = \frac{NN^2{}_i}{NN_{SUM}} \tag{5}$$

Every node uses neighbor level to decide if it is a critical node or not. If a node detects that its neighbor level is lower than a pre-specified threshold, it regards itself as a critical node. A critical node will perform the *backup operation* to send to all connected neighbors the *backup message* containing the connected neighbor list. When a node j receives a backup message, it will check the neighbor list attached with the message. If there are any nodes not in j's neighbor list, node j would store them in the *stock neighbor list*. Stock neighbor list is different from the normal neighbor list. It keeps some spare nodes which can be used to avoid overlay partition. The capacity of stock neighbor list is bounded. When the stock neighbor list is full, FIFO (first-in-first-out) rule is used to replace nodes in the list.

A node i in the system periodically sends a *check message* to all nodes in the stock neighbor list to request their connected neighbor lists. When a node j receives the check message, it would reply node i with its own neighbor list. After receiving the neighbor list from node j, node i does nothing if: (a) none of the nodes in j's neighbor list is within i's AOI or (b) all nodes in j's neighbor list have already known by i. Otherwise, node i will add into its neighbor list the additional nodes that is not in i's neighbor list but is within i's AOI. Furthermore, node i will connect to the additional nodes immediately. By doing the operations mentioned above, potential overlay partition may be avoided.

5 Simulation Results

In this section, we perform simulations to show the negative impacts of node mobility and node failure on neighborship consistency for VON-based DVEs. We also perform simulations to show that the proposed methods in the last section can effectively reduce the impact.

The simulation environment is assumed to have 1000 nodes in a 1200-unit by 1200-unit plane, where nodes move randomly with a constant velocity of 5 units per time-step. The underlying overlay is VON and the initial AOI radius is 100 units. The number of directly connected neighbors is limited to be 20. We assume that dynamic AOI is adopted. That is, if the number of neighbors of a node exceeds 20, the node will decrease its AOI radius until it has less than or equal to 20 neighbors; otherwise, the node will increase its AOI radius up to 100 units.

5.1 Simulation for Node Mobility

In this subsection, we perform simulation to show the effect of node mobility on neighborship consistency. We use *relative speed*, the ratio of speed to AOI radius, as the assessment of mobility. We define *relative speed (RS)* as:

$$RS = \frac{Speed}{AOI\ radius} \tag{6}$$

In equation (6), Speed is measured by units per time step and AOI radius is measured by units. In simulation, all nodes have the same RS which is of the value from 5% to 50%. As shown in Figure 5, neighborship consistency decreases when RS increases. When the RS is 40% or 50%, neighborship consistency is under 30%. Only when the RS is under 10%, neighborship consistency can be over 90%.

In adaptive AOI buffer mechanism, we set AOI buffer width to be 1RS or 1.5RS, where 1RS and 1.5RS mean that we use 1 and 1.5 multiply current relative speed as the AOI buffer width. For example, if relative speed of a node is 5%, 1RS and 1.5RS mean that the AOI buffer widths are 5% and 7.5% of the AOI radius. In Figure 5, we can see that the adaptive AOI buffer mechanism with 1RS and 1.5RS both improve neighborship consistency. However, by Figure 6, we can see that the mechanism also incurs higher control overhead (i.e., the total number of bytes transmitted per node gets larger) than original VON design. According to the simulation results, we conclude that there is a tradeoff between control overhead and neighborship consistency. If we pursue better neighborship consistency, then we should spend more control cost.

5.2 Simulation for Node Failure

In this subsection, we perform simulation for the effect of node failure. The spread of nodes in the system is of random or clustered distributions. For random distribution, nodes spread uniformly in the DVE plane. In clustered distribution, three attractors are assumed to be located randomly in the plane to attract nodes to move towards them. Nodes will gather around the attractors. We want to find proper parameter values to achieve better neighborship consistency with affordable control overhead. The parameters investigated are (a) the size of stock neighbor list, (b) the checking period, and (c) the discovery period.

We first fix parameters (b) and (c) to 10 and run simulation with varying parameter (a) values and fixed node failure rate 30%. In this way, we obtain a proper value for parameter (a). And then we can also obtain the proper values for parameters (b) and (c) in similar ways. By the simulation results in Figure 7, we can see that the proper value of parameter (a) is 20 because it makes the average number of partitions close to 1 and the control overhead is affordable.

We can see the proper value of parameter (b) is 10 by observing Figure 8 because such a parameter setting makes the average number of partitions close to 1 and makes the control overhead affordable. Similarly, we can conclude that the proper value of parameter (c) is 10 by the results shown in Figure 9.

After obtaining proper values for the parameters (a), (b) and (c), we use these values in the following simulation for the critical node mechanism.

We perform the simulation with 10%, 20% and 30% node failure rates. Every simulation lasts 2000 steps; it first goes 300 steps normally and then 10% (or 20%, or 30%) of nodes are set to fail simultaneously. Figure 10 and Figure 11 are for the results of the clustered distribution simulation, in which the nodes cluster into three groups in the first 300 steps and can move randomly after node starts to fail. Figure 12 and Figure 13 are for the random distribution simulation, in which nodes can move randomly in all simulation steps.

By observing Figures 10 and 11, we can see that node failure indeed has negative impact on neighborship consistency for original VON design under clustered distribution. The neighborship consistency drops to around 80% and the number of partitions grows to around 20 when 30% of nodes may fail simultaneously. We can also see that the critical node mechanism can reduce the negative impact of node failure dramatically. After applying the critical node mechanism, the neighborship consistency is still near 100% and the number of partitions is kept to be 1, 2 or 3 for failure rates of 10%, 20% and 30%.

By observing Figures 12 and 13, we can see that node failure has not-so-significant impact on neighborship consistency for original VON design under random distribution. The neighborship consistency is still around 99% and the number of partitions is 1 or 2 when 10% of nodes may fail simultaneously. However, the neighborship consistency goes down to 88% and the number of partitions is around 10 when 30% of nodes may fail simultaneously. The critical node mechanism can also help reduce the negative impact of node failure. The neighborship consistency is around 93% and the number of partitions is around 7 when 30% of nodes may fail simultaneously. Nevertheless, the critical node mechanism has less improvement under the random distribution model than under the clustered distribution. This is because there are few nodes detecting themselves to be critical nodes under random distribution.

We have seen that the critical node mechanism can reduce the negative impact on neighborship consistency of node failure. Below, we investigate the overhead introduced by the mechanism. By Figure 14, we can see that the control overhead of applying the critical node mechanism is just a little more than that of the original VON design.

6 Conclusion

In this paper, we focus on maintaining good neighborship consistency for P2P DVEs, especially for VON-based DVEs. We have identified two factors that affect neighborship consistency and proposed mechanisms to reduce their negative impact. The first factor is node mobility. We propose the mechanism of adaptive AOI buffer to reduce its influence on neighborship consistency. The second factor is node failure. We show that simultaneously failing nodes may lead to overlay partition, which will do much harm to neighborship consistency. We propose the mechanism of critical node detection to avoid such a situation. We have also performed intensive simulation experiments to demonstrate the improvement of the proposed mechanisms. By the simulation results, we conclude that the proposed mechanisms can improve neighborship consistency substantially, while the control overhead is kept low.

References

[A91] F. Aurenhammer, “Voronoi diagrams—a survey of a fundamental geometric data structure,” *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 345-405, 1991.

[DG99] 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.

[HCC06] Shun-Yun Hu, Jui-Fa Chen and Tsu-Han Chen, "VON: A scalable peer-to-peer network for virtual environments," *IEEE Network*, vol. 20 no. 4, pp. 22-31, 2006.

[KAM04] 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, pp. 353–370, 2004.

[KS03] J. Keller and G. Simon, “Solipsis: A massively multi-participant virtual world,” in *Proc. Int. Conf. Parallel and Dist. Tech. and App. (PDPTA 03)*, 2003, pp. 262-268.

[MKMA04] N. Matsumoto, Y. Kawahara, H. Morikawa, and T. Aoyama, "A Scalable and Low Delay Communication Scheme for Networked Virtual Environments," in Proceedings of 1st IEEE International Workshop on Networking Issuess in Multimedia Entertainment (NIME2004), Dallas, TX, U.S.A, November 2004.

[MT95] D. C. Miller and J. A. Thorpe, “SIMNET: The Advent of Simulator Networking,” in *Proc. IEEE*, vol. 83, no 8, pp. 1114-1123, Aug. 1995.

[SKH01] J. Smed, T. Kaukoranta, and H. Hakonen, "Aspects of Networking in Multiplayer Computer Games," in *Proc. of Int. Conf. on Application and Development of Computer Games (ADCOG)*. Nov. 2001, pp. 74-81.

[SMKKB01] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” in *Proceedings of SIGCOMM2001*, Aug. 2001, pp. 149–160.

[SZ99] S. Singhal and M. Zyda, *Networked Virtual Environments: Design and Implementation*, ACM Press, New York, 1999.

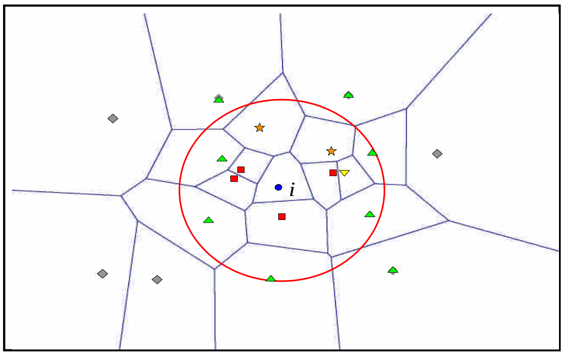

Figure 1. Different types of neighbors of node *i* (the circle (●) stands for node *i* and the big circle is its AOI; squares (■) are its enclosing neighbors; triangles (▲), boundary neighbors; stars (★), both enclosing and boundary neighbors. The inverted triangles (▼) are normal AOI neighbors, and diamonds (◆) are irrelevant nodes.).

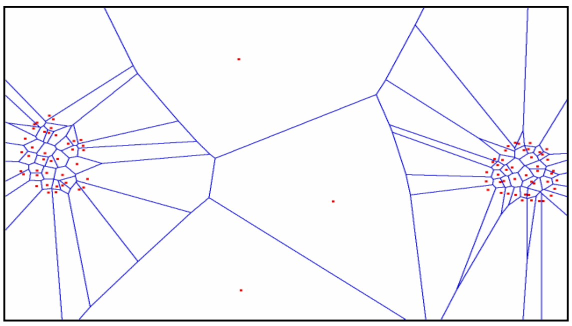

Figure 2. An example of VON-based DVE plane in which overlay partition may occur

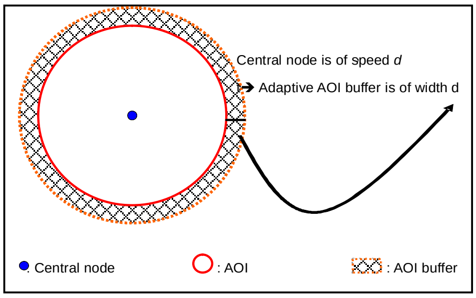

Figure 3. An example of adaptive AOI buffer

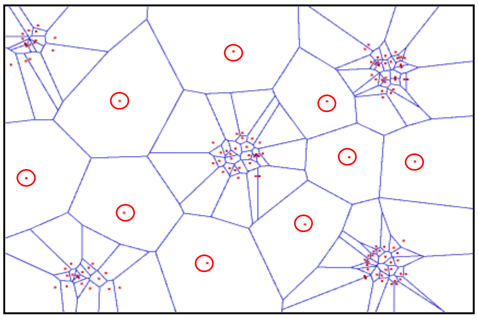

Figure 4. An example of critical nodes

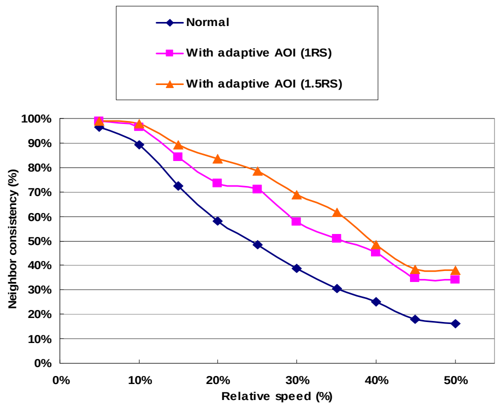

Figure 5. Effect of node mobility for VON-based DVEs

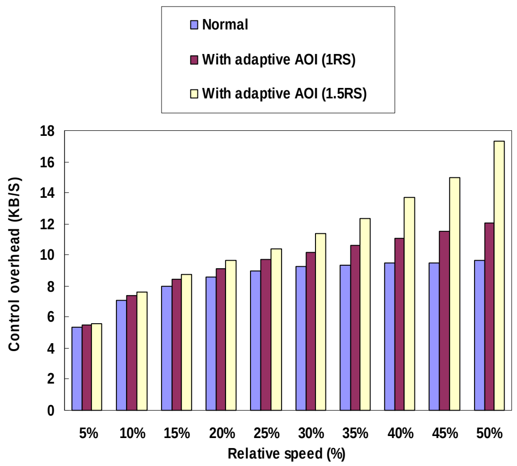

Figure 6. Control overhead of adaptive AOI buffer mechanism

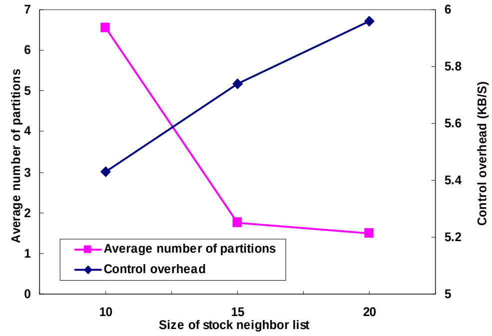

Figure 7. Simulation results for different stock neighbor list sizes

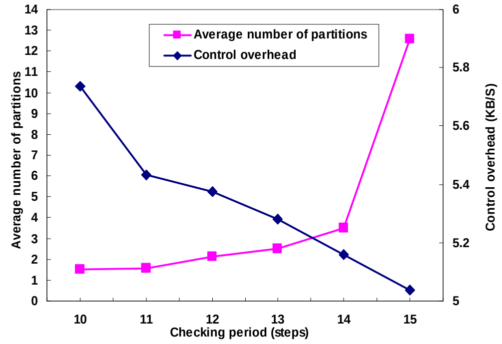

Figure 8. Simulation results for different checking periods

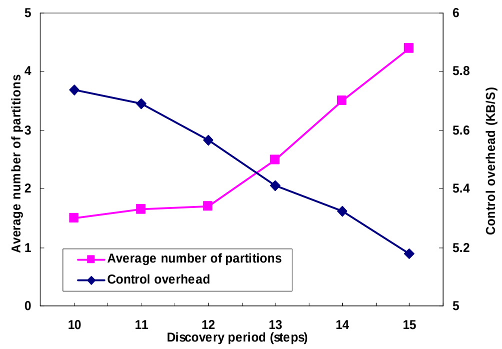

Figure 9. Simulation results for different discovery periods

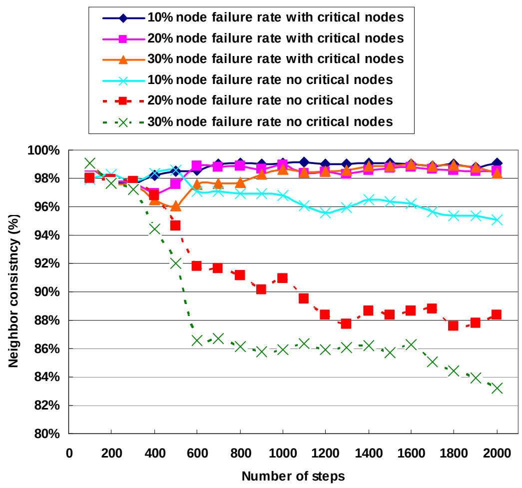

Figure 10. Neighbor consistency under different node failure rates for clustered distribution.

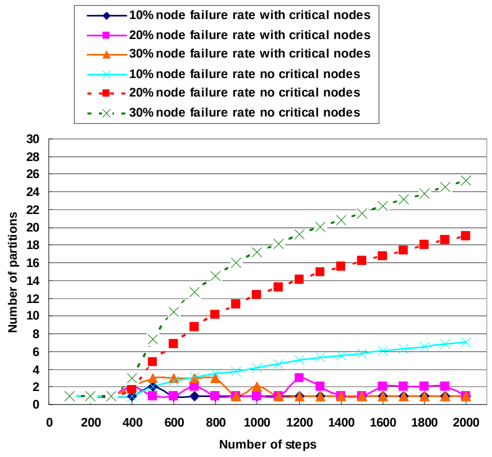

Figure 11. The number of partitions under different node failure rates for clustered distribution.

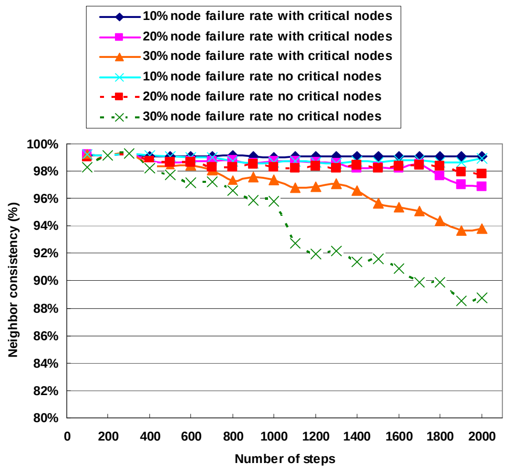

Figure 12. Neighbor consistency under different node failure rates for random distribution.

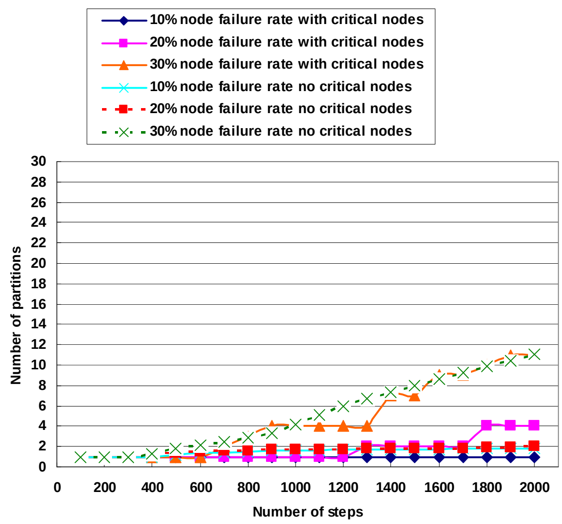

Figure 13. The number of partitions under different node failure rates for random distribution

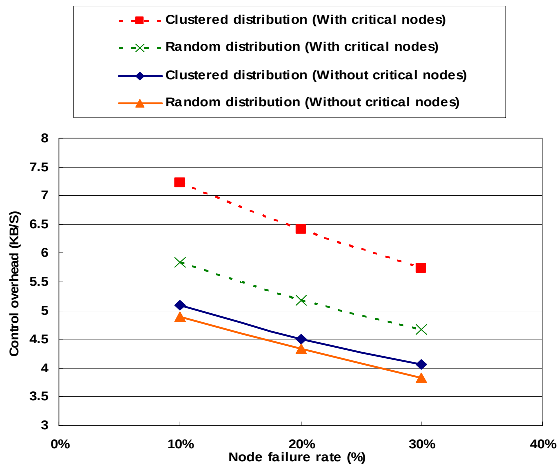

Figure 14. Control overhead with and without the critical node mechanism