Application Layer Multicast with Proactive Route Maintenance over Redundant Overlay Trees

Yohei Kunichika, Jiro Katto and Sakae Okubo

Department of Computer Science, Waseda University
{yohei , katto}@katto.comm.waseda.ac.jp , sokubo@waseda.jp

***Abstract*— In this paper, an efficient algorithm to look for backup parents in preparation of parent leaving is proposed for application layer multicasting whose topology is constituted in the shape of a tree from a single source node. In most conventional methods, each child node starts searching for its new parent after its parent node leaves from a multicasting tree. This reactive operation often causes long interruption period. In our proposal, each node holds its parent candidate proactively over redundant overlay trees. Proactive route maintenance leads to smooth switching to a new parent after node leaving and failure, and redundant structure of a multicasting tree avoids exhausting search of a backup parent. Computer simulations are also carried out and effectiveness of the proposed approach is verified.**

***Index Terms*— Application Layer Multicast, P2P Streaming, Proactive Route Maintenance**

I. Introduction

Internet broadcasting has attracted attention of many researchers since the advent of IP multicasting [1]. The IP multicasting is an effective mechanism that can completely eliminate redundant data delivery to the multiple subscribers. However, the IP multicasting suffers from its quite slow deployment in the Internet due to inefficient support of native IP multicasting by the routers of current commercial ISPs. Application layer multicast (ALM) or overlay multicast emerges as an alternative to the IP multicasting. It enables packet multicasting delivery in an application layer without changing any network infrastructure of the current Internet. Instead of extending router functions, each end host receives a packet, replicates and forwards it to the next end hosts on an overlay network.

The most active research area about the ALM is a design of routing protocols [2]-[13]. There are several measures to evaluate effectiveness of the routing protocols as follows: (a) *quality of the data delivery path*, that is measured by stress, stretch and node degree parameters of the overlay multicast tree against native IP multicasting, (b) *robustness of the overlay*, that is measured by the recovery time to restore a packet delivery tree after abrupt end host failures, and (c) *control overhead*, that represents scalability of the protocols against large number of receivers.

One of the unavoidable problems of the ALM is that end hosts have to reconstruct the overlay network after a node leaves the multicast session or fails. In IP multicast, because the non-leaf nodes in the delivery tree are routers, we do not have to take into account the preceding problems. However, in the ALM, the non-leaf nodes are end hosts. End hosts are free to leave the multicast session, hence it is important to restore the packet delivery tree in these cases. Nevertheless, although quite a lot of routing protocols are proposed, most researchers focused on reactive restoration of a delivery tree. That is, end hosts start to search for its new parent after its old parent node departures.

Some researchers considered proactive approaches, in which backup routes are maintained before the parent departure happens. In Probabilistic Resilient Multicast (PRM) [12], each host chooses a constant number of other hosts at random and forwards data to each of them with a low probability. This operation indirectly contributes to backup route maintenance. However, PRM generates extra data overheads that cannot be negligible for some applications such as live media streaming.

Another proactive approach called Yang's approach in this paper employed control packets instead of data packets [13]. In this approach, backup routes are calculated proactively whenever a node leaves or joins. When a node leaves, a backup route previously calculated is immediately applied and next backup routes are updated. When a new node joins, next backup routes are calculated without activating the previous backup route. Calculation of the backup routes is lead by two nodes; the parent node and the grandparent node of the leaving/joining node. A problem of this approach is that, when a node cannot form a backup route due to degree constraints of its children nodes, it employs grandchildren and below until a

node without the degree constraint will be found. Another problem is that two nodes have to be involved in the backup route calculation whenever leave/join events happen.

We therefore propose a novel proactive approach that avoids the degree limitation problem by forcing each node to prepare a redundant route for backup. Each host communicates with its grandparent and registers its backup parent information. When a node leaves the overlay network, it is guaranteed that all of its children can connect to their backup parents at worst in the same layer. In addition, the number of nodes to be engaged in the backup route update will be reduced to unity as shown later.

The rest of this paper is organized as follows. Section B reviews the related work. Section C proposes the proactive backup route maintenance mechanism. Section D provides results of performance evaluation. Finally, Section E concludes this paper.

II. Related work

Most of the application layer multicast protocols focus on how to construct an efficient multicast tree. Narada [2] and ALMI [3] are mesh-first protocols that were designed for small groups. Scattercast [4] is another mesh-first protocol for larger groups, which utilizes a set of proxies to which end hosts register. Yoid [5], Overcast [6] and Peercast [7] are tree-first protocols for larger groups. Bayeux [8] and CAN-based multicast [9] utilize structured P2P routing known as the distributed hash table (DHT) algorithm. ZIGZAG [10] and OMNI [11] are recently reported ALM protocols. ZIGZAG organizes a multi-layer hierarchy of bounded-size clusters of peers and constructs a multicast tree for each peer to receive contents from a "foreign" head. This procedure avoids occurrence of network bottleneck and keeps small end-to-end delay. OMNI presents a decentralized and adaptive update mechanism of an overlay network to minimize average-latency to the entire hosts with degree constraints. However, for node leaving and failures, these protocols employ reactive actions, i.e. they start to find new routes after the node departure happens. Therefore, long interruption period sometimes occurs in children nodes, which severely degrades streaming performance.

On the other hand, PRM [12] uses a proactive approach of reconstructing overlay network. Randomized forwarding seems to be effective in some situation, but this scheme may generate huge overhead traffic of backup route maintenance. Yang's approach [13] forces each non-leaf host to pre-calculate a backup parent for each of its children. In this scheme, recovery time to restore overlay network is cut down with appropriate amount of control packets. However, in this approach, pre-calculation of a backup route sometimes consumes heavy computational cost because degrees of upper layer nodes are usually filled up. A node that seated in upper layers in the overlay tree cannot find its backup route until going down to lower layer nodes having no degree constraint.

III. Proposed Method

Construction of an overlay network is similar to constructing a degree-constrained spanning tree. Each end host is restricted to have children that receive data from the host, because the bandwidth is limited between the host and its children. It is known that the degree-constrained minimum spanning tree problem is an NP-complete problem, and many researchers had tried to alleviate the problem with some heuristics. However, instead of focusing on this problem, this paper pays attention to how to reconstruct a feasible spanning tree that can recover quickly in the case of node leaving and failures.

A. Reactive Reconstruction of an Overlay Network

When a host "leaves" an overlay network, it can send a message to inform affected nodes of its leaving the network. When a host suddenly "fails", it cannot send a message, so affected nodes have to detect this failure by some sort of heartbeat mechanism.

We first take up Peercast [7] as an example of the reactive approaches. It proposed some recovery processes after a node leaves; called *root*, *root-all*, *grandfather*, and *grandfather-all*. In the root mechanism, when a node leaves the network or fails, each of its child contacts to the source node to rejoin the network. A subtree rooted at each child does not change. Root-all is a mechanism in which, when a node leaves or fails, all its descendant contact to the source node. Grandfather and grandfather-all mechanisms are almost same as the previous mechanism except for the contacted node. In Grandfather and Grandfather-all policies, the nodes of which parent leaves try to contact to their grandfather. Similar to Root and Root-all mechanisms, contacting nodes are the children of the leaving node in Grandfather policy. On the other hand, all the descendant nodes of the leaving node will contact in Grandfather-all policy. Among these, we choose the "grandfather" policy for comparison purpose because its performance is most graceful overall. In this grandfather policy, when a node leaves, the children of the leaving node contact their grandfather. If a node fails, the children of the node contact the source node rooted at the overlay tree because they cannot recognize their grandfather. The main task is to find a new parent for each affected child as quickly as possible. However, especially in the node failure phase, it takes longer time because each affected node searches for its new parent by contacting to the rooted node in the tree, that might be quite far from the affected node. Furthermore, when the upper layer nodes are filled up in degrees, backup parent finding operation has to be repeated to reach the node that might be located at the lower layer (possibly, a leaf node).

B. Proactive Route Maintenance over Redundant Overlay Trees

We therefore apply a proactive approach in order to reduce the time of restoration of an overlay network. It is most important that we construct an overlay tree without each host

maximizing its out-degree.

Total out-degree may be calculated by the bandwidth of the connection of an end host divided by the media playback rate. If an end host has total out-degree = n, it can have n children.

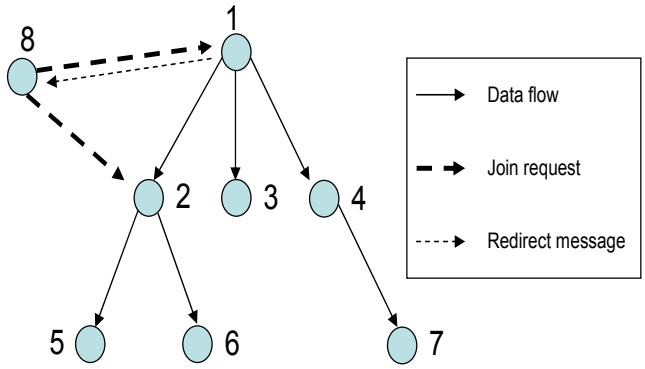

Fig. 1. An example of new node participation.

However, in our scheme, we force an end host with total out-degree = n to have n-1 children only. This limitation simplifies backup route calculation, in which parent finding operation is completed at the children layer, and contributes to overhead reduction. Let us explain an example in detail.

Firstly, new node participation process is carried out as follows. In Fig.1, it is assumed that total out-degree of each node is equal to 4. In the previous work, when new node 8 requests to connect to node 1, node 1 accepts node 8 as its child because its degree is not filled. However, in our proposal, node 1 refuses the request because the rest of degree of node 1 is only one. As a result, node 8 is redirected to node 2.

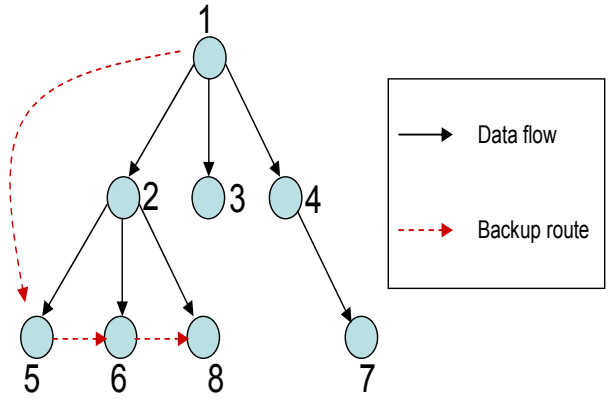

Fig. 2. An example of backup route selection.

Next, backup route calculation is carried out by the parent node as follows. In Fig.2, when node 8 connects to node 2, node 2 checks its children list. Since node 2 has three children, when node 2 leaves, node 1 cannot accommodate all the children of node 2 due to its degree constraint. Therefore, node 2 sends the children list to node 1. Node 1 then measures a round trip time to each grandchild, and informs the fastest node that node 1 will be its backup parent. Node 1 also informs the other nodes that faster nodes will be their backup parents. In Fig.2, node 8 is the fastest, node 5 is the second and node 6 is the last. Therefore, node 8 chooses its grandparent, node 1, node 5 chooses node 8, and node 6 chooses node 5, respectively, as their backup parents. As a special case, when the children list of node 2 includes node 8 only (i.e. no other children exist), node 2 immediately informs node 8 that node 1 will be a backup parent of node 8.

This backup route calculation is also carried out whenever the node leaving/failure event happens similar to the node participation case. When a node leaves a network, the backup route which skips the vanished node is immediately applied and the new backup route calculation follows. Note that layers to which the backup route calculation is applied are limited at worst at the grandchild layer. It never goes down to the lower layers dissimilar to the previous approaches.

Correspondence when two or more members leave simultaneously is shown in [13]. It can respond to this situation since the number of hosts holding the backup route to a higher layer is only one.

Backup routes created above are certainly efficient as long as each host does not fully utilize its out-degree. However, it is possible that a host maximizes its out-degree by accommodating a new node after restoring an overlay tree. When this happens, a tree reconstruction procedure is invoked by the host itself in order to recover the route redundancy. Currently, this procedure is carried out by asking the children and below except the newly connected node whether their out-degrees are filled up. When an acceptable node is found, the newly connected node is moved to the acceptable node. Fig3 shows this case. Node 2 maximizes its out-degree because node 8 has joined. Node 2 asks its children except node 8 whether it can connect to new node. Then nodes 5, 6, and 7 send hit messages to node 8 if each of them can receive node 8. In this case, node 8 receives the first message from node 6, so node 8 joins node 6.

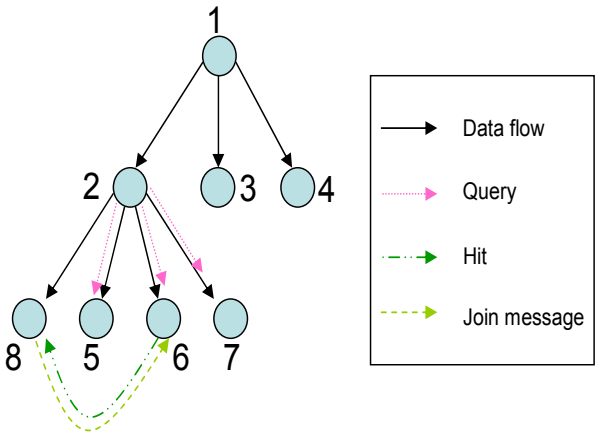

Fig. 3. An example of redundant tree reconstruction.

IV. Performance Evaluation

We carry out computer simulations using ns-2 simulator [14]. We are mainly interested in the resilient performance that indicates how fast the overlay tree can be restored and in the reduction of control overheads thanks to provision of redundant routes. We compare our scheme with the promising reactive scheme, called the grandfather policy, described in subsection B.1. We also compare our scheme with Yang's approach, which is the previous proactive scheme proposed in [13].

Our simulation topology has 24 routers, in which four of them are domain-to-domain routers. End hosts randomly connect to one of the 20 routers except the four inter domain routers. Fig. 4 represents a part of our topology. The number of hosts varies from 50 to 400. The link latencies vary from 10 ms to 100ms. The out-degree of each host is fixed at 4. The overlay tree is constructed at once after each experiment starts. Then end-hosts randomly join and leave the tree every 10 seconds.

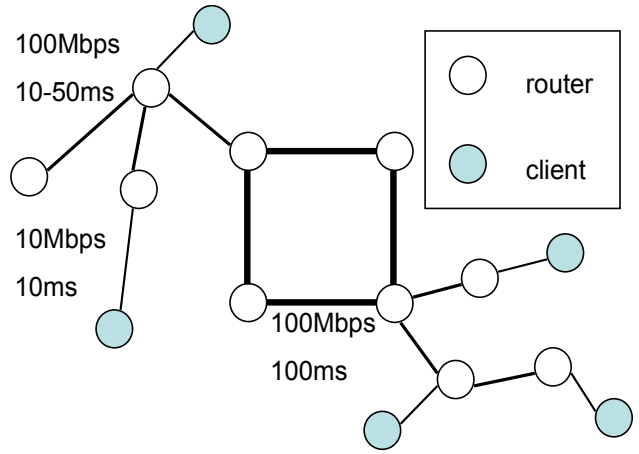

Fig. 4. Network topology used in computer simulations.

A. Comparison of Recovery Latencies

Fig. 5 compares recovery latencies of grandparent policy of the reactive approach, Yang's approach and the proposed backup route maintenance method. Recovery latency is the time after an affected node of a leaving node connects to a new parent and receives data packets from the parent. From this figure, we can recognize that the average recovery time of the reactive method is twice or more higher than that of the proactive methods when a node leaves, and 10 times higher when a node fails. This result is relevant because it is almost proportional to the average number of nodes contacted by children nodes of the leaving node. The proactive methods enable the affected nodes to connect to their backup parents immediately. This is common in both proactive methods, so their results are nearly equal. On the contrary, in the reactive method, the request may be rejected by the contacted node due to the degree constraint and be repeated until it will be accepted. Probability of this rejection becomes higher when each node contacts to an upper layer node, especially in the node failure case, where affected nodes have to contact to the source node rooted at the overlay tree. As the number of end-hosts increases from 50 to 400, the recovery time of the reactive approach for node failure increases. This is because the height of an overlay tree becomes deeper on the average. On the other hand, recovery latencies of the proactive approaches are almost the same independently of the number of nodes because it is enough for each affected node to contact only one node.

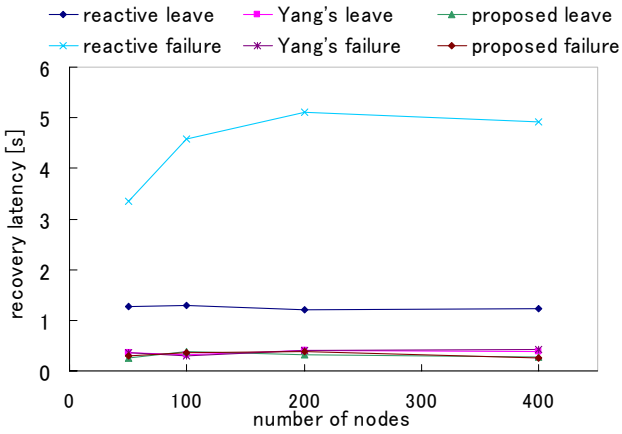

Fig. 5. Comparison of recovery latencies when node leave or failure happens.

B. Comparison of Control Overheads

Fig. 6 compares total number of control packets of the reactive method, Yang's method and the proposed method. Control packets represent all signaling packets except data packets and heartbeat messages. From this figure, we can recognize that Yang's approach generates higher control packets than others. Fig. 7 shows particulars (*join*, *redirect*,

search, *leave* and *backup*) about control packets when the number of the nodes is 200. In addition to the possible iterative search problem of Yang's method, another disadvantage is that it has to activate two nodes (parent and grandparent) for its backup route maintenance, because addition of a new node (child) sometimes invalidates previous backup routes that have to be updated by the grandparent. Note that our scheme activates a parent node only due to the route redundancy. This contributes to drastic reduction of control packets for backup routes as shown in the figure. We also observe that our scheme performs almost similar to the reactive scheme. This is because the reactive method generates more packets when iterative parent search is invoked. We can recognize that there are a large number of control packets for join, redirect and search in reactive method. Thanks to the route redundancy again, our approach does not cause the iterative requests even if it generates excessive control packets for backup route maintenance. As a result, both overheads are almost the same.

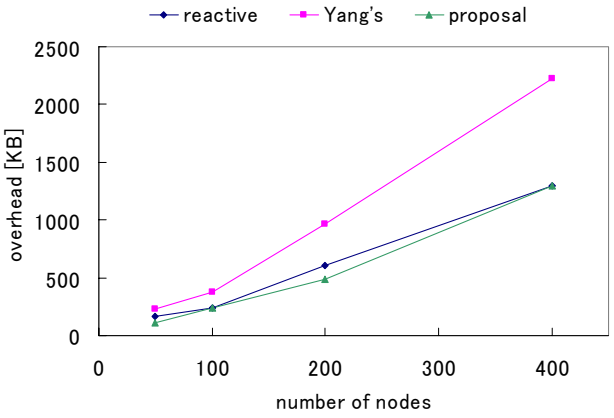

Fig. 6. Comparison of control overheads according to the number of nodes [control packet size = 128 byte].

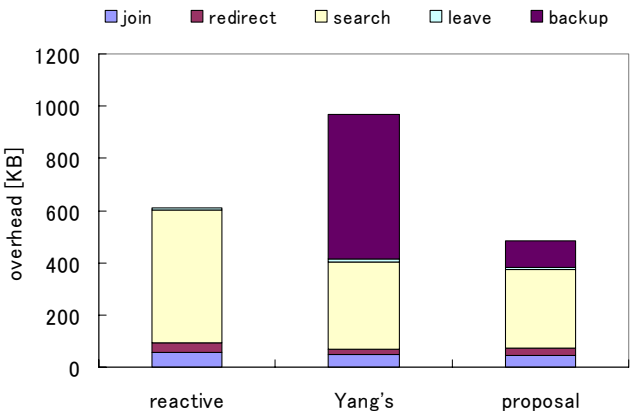

Fig. 7. Detailed comparison of control overheads when the number of nodes is 200 [control packet size = 128 byte].

C. Comparison of Data Delivery Delays

1) Pilot Studies

Proposed redundant tree simplifies a backup route search and contributes to overhead reduction. However, the redundancy causes an overlay tree to be longer and possibly leads to overall delay increase.

A node located deeper in a tree is likely to receive the same data packets later than the shallower ones because the hop counts from the source to lower layer nodes monotonically increases. Therefore, an obvious problem of our approach is possible increase in data delivery delays, especially when the node degree is small or the number of nodes increases. However, this disadvantage is expected to be negligible when the node degree is sufficiently large. Larger degree leads to smaller depth of a multicast tree even if one connection is reserved for backup route. Next two results support these considerations.

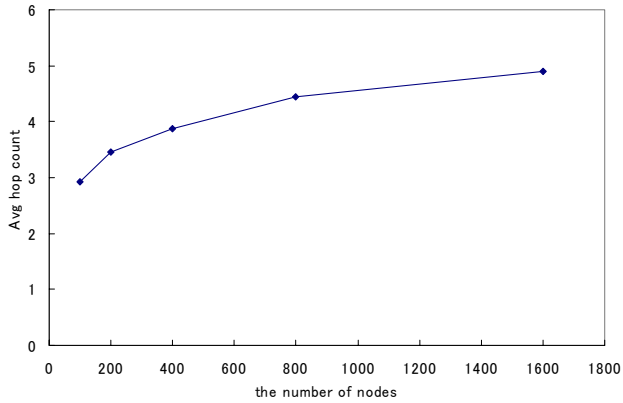

Fig. 9. Relationship between average hop counts and the number of nodes, where the node degree is four.

Fig. 9 shows relationship between the average hop count and the number of nodes of which degree is equal to four. This hop count means hop count in an overlay network. It corresponds to the depth of a tree, and it is expected that average delay will increase as the number of nodes increase.

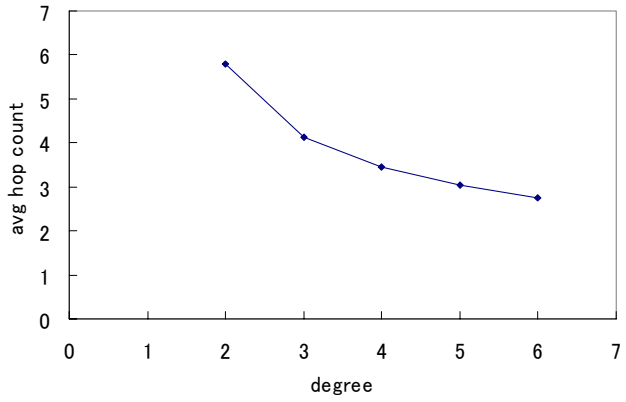

Fig. 10. Relationship between average hop count and node degree, where the number of nodes is 200.

Fig. 10 shows relationship between the average hop count and node degree when the number of nodes is 200. This figure suggests that the hop count is monotonically decreasing as the node degree increases. Therefore, it is expected that average delay will decrease as the node degree increases.

2) *Simulation Results*

Previous studies suggest that our proposal is likely to make an overlay tree deeper and to increase the data delivery delays. However, this influence can be eased when the node degree increases. Based on these expectations, we carry out next simulation.

Fig. 11 compares the data delivery delay when varying the node degree from three to six, where the number of nodes is fixed at 200 in all methods. In this figure, *Avg* represents the average delay of each node, and Max represents the worst delay of all nodes. Essentially Our proposal shows the largest delay. However, we can recognize that, as the node degree becomes larger, the difference between our proposal and the others becomes smaller. As stated before, the reason is that larger node degree contributes to smaller height of a tree, which leads to reduction of delays. We can see that the delays of all methods are almost the same when the node degree is six in the case.

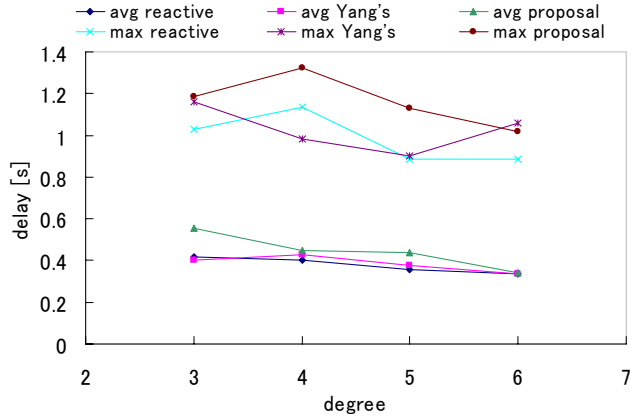

Fig. 13. Comparison of data delivery delays when changing the degree of a multicast tree, where the number of node is 200.

V. Discussion

Some problems still remain in our proposal to be solved.

Firstly, we assume all hosts have the same degree values in our experiment. However, in actual environment, they are different and some hosts might accept none nodes. To solve this heterogeneity, we are preparing to improve our proposal. In presence, all hosts construct an overlay network with one node degree left. However in modified version, hosts used up the degree or leave it depending on the situation. Secondly, we will extend experiment environments. We try to experiment in some different topologies such as large scale situation. Finally, we are currently preparing to experiment in actual network utilizing C++ and socket programming.

VI. Conclusions

This paper proposed proactive backup route maintenance over redundant overlay trees in order to enable smooth tree recovery and to reduce control overheads. Computer simulations were carried out, and it was verified that the recovery latencies were drastically reduced while the overhead of control packets was almost the same as that of the reactive approach. Although the data delivery delay was larger than other methods, the influence is alleviated when the degree is large. Furthermore, improvement to the existing proactive approach is also provided quantitatively. As future work, we will try to something as stated previously.

References

[1] S. Deering, "Host Extension for IP Multicasting," RFC 1112, Aug.1989.
[2] Y. Chu, S. G. Rao, H. Zhang, "A Case for End System Multicast," in *Proceedings of ACM SIGMETRICS 200*0, June. (2000)
[3] D. Pendarakis, S. Shi, D. Verma, M. Waldvogel, "ALMI: An Application Level Multicast Infrastructure," *3rd USENIX Symposium on Internet Technologies and Systems*, Mar. (2001)
[4] Y. Chawathe, S. McCanne, E. Brewer, "Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service," PhD Thesis, University of California, Berkeley, (2000)

[5] P. Francis, "Yoid: Extending the Internet Multicast Architectuire," http://www.icir.org/yoid/
[6] J. Jannotti, D. Gifford, K. Johonson, M. Kaashoek, J. O'Toole, "Overcast: Reliable Multicasting with an Overlay Network," *4th Symposium on Operating Systems Design & Implementation*, Oct. (2000).
[7] H. Deshpande, M. Bawa, H. Garcia-Molina, "Streaming Live Media over Peers," Technical Report 2002-21, Stanford University, Mar. (2002)
[8] S. Zhuang, B. Zhao, A. Joseph, R. Katz, S. Shenker, "Bayeux: An Architecture for Scalable and Fault-Tolerant Wide-Area Data Dissemination," *ACM NOSSDAV 2001*, June. (2001)
[9] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Application-level Multicast using Content-Addressable Networks," In Proceedings of NGC (2001)
[10] D. Tran, K. Hua, T. Do, "ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming," in *proceedings of IEEE INFOCOM 2003*, Apr. (2003)
[11] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, S. Khuller, "Construction of an Efficient Overlay Multicast Infrastructure for Real-time Applications," in *proceedings of IEEE INFOCOM 2003*, Apr. (2003)
[12] S. Banerjee, S. Lee, B. Bhattacharjee, A. Srinivasan, "Resilient multicast using overlays," in *proceedings of ACM SIGMETRICS 2003*, June. (2003)
[13] M. Yang, Z. Fei, "A Proactive Approach to Reconstructing Overlay Multicast Trees," in *proceedings of INFOCOM 2004*, March. (2004)
[14] The Network Simulator –ns-2, http://www.isi.edu/nsnam/ns