Multicast Tree to Minimize Maximum Delay in Dynamic Overlay Network

Chae Y. Lee and Jin woo Baek
Dept. of Industrial Engineering, KAIST,
373-1 Kusung Dong, Taejon, Korea
{chae, bjw77}@kaist.ac.kr

Abstract

Overlay multicast technique is an effective way as an alternative to IP multicast. Traditional IP multicast is not widely deployed because of the complexity of IP multicast technology and lack of application. But overlay multicast can be easily deployed by effectively reducing complexity of network routers. Because overlay multicast resides on top of densely connected IP network,

In case of multimedia streaming service over overlay multicast tree, real-time data is sensitive to end-to-end delay. Therefore, moderate algorithm's development to this network environment is very important.

In this paper, we are interested in minimizing maximum end-to-end delay in overlay multicast tree. The problem is formulated as a degree-bounded minimum delay spanning tree, which is a problem well-known as NP-hard. We develop tabu search heuristic with intensification and diversification strategies. Robust experimental results show that is comparable to the optimal solution and applicable in real time

Keywords

Overlay multicast, Multicast tree, Tabu search

1. Introduction

Overlay network has become an effective way to provide advanced services in the Internet even without making any changes in the network layer. Overlay network is formed by a subset of underlying physical nodes. The connection between the overlay nodes is provided by overlay links, each of which is usually composed of one or more physical links [13]. One advanced service on overlay network is multicast.

Multicast is an important part of many next generation applications, including voice conferencing, video-on-demand and peer-to-peer file sharing. Multicast is an efficient transmission method that simultaneously provides a same type of service to a group of receivers. Router plays a significant role of replicating data packets and transmitting them to receivers, so that the source would not have to transmit the same data repeatedly. However, the deployment of multicast has not been taken by most commercial ISPs. Its deployment is difficult due to the complexity of technology and lack of applications.

Overlay multicast is highlighted as an alternative for providing multicast services in the Internet [3, 4, 5, 6, 7, 12, 17, 18]. In overlay multicast, data packets are replicated at end hosts. Each multicast member is responsible for forwarding the packets. Thus, overlay multicast does not change the network infrastructure but performs forwarding at each end host. The basic idea of multicast and overlay multicast is described in Figure 1. In Figure 1 (c), Logical tree structure is built to form an overlay multicast network regardless of the underlying physical Internet.

Most of the previous work on overlay multicast can be classified into two broad categories [16, 17]. One is end-to-end overlay and the other is proxy-based overlay. In end-to-end overlay, the end nodes are logically connected together to form an overlay network. These end nodes have the responsibility to forward data to other nodes. Proxy-based overlay is a hierarchical structure compared to end-to-end overlay. This overlay uses fixed nodes as proxies to facilitate overlay services. Most of these fixed nodes are deployed in order to achieve similar functions, such as routing and forwarding. Multicasting service is performed with the help of proxy nodes, which can duplicate data and forward them to end hosts with predefined routing algorithm. These two overlay networks are depicted in Figure 2.

In end-to-end overlay multicast, each multicast member node has the responsibility of forwarding packets. Therefore, when the multicast member nodes leave or join a multicast session, overlay multicast tree should be recovered. Such phenomenon can be explained by members logging off and disconnecting themselves from the multicast session or new members subscribing for the multicast service.

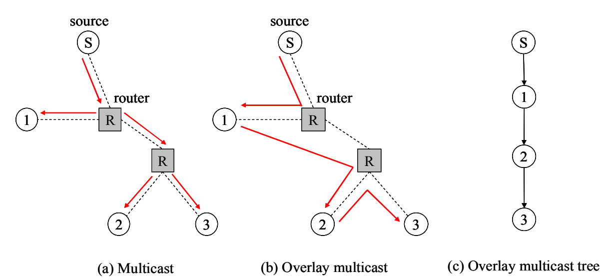

Figure 1. Overlay Multicast

In this paper, we are interested in minimizing maximum end-to-end delay in overlay multicast tree. Tabu search heuristic is developed with intensification and diversification. The rest of the paper is organized as follows. In section 2, we discuss other subjects related to our work. In section 3, we describe our problem in detail. In section 4, we present a heuristic algorithm to solve the recovery of multicast tree. In section 5, the performance of suggested algorithms is analyzed. Finally, we present our conclusion in section 6.

2. Related work

There are a number of studies on overlay multicast services in the recent literature, mostly due to the efficiency of overlay network. Some studies dealt with similar subjects to our paper.

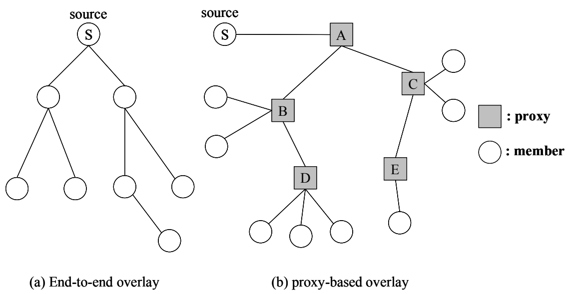

Figure 2. Comparison of overlay multicast tree

In [17], proxy-based architecture is applied. This paper focuses on minimizing average-latency to all the end hosts. It presents a decentralized scheme that organizes the Multicast Service Nodes (MSNs) into an appropriate overlay structure that is particularly beneficial for real-time applications. However, our paper tries to minimize maximum overlay delay. As a measure, minimizing the maximum delay is fairer than minimizing the average delay. We formulated this problem and suggested a tabu search algorithm.

In [15], to cope with the node failure in the overlay multicast network, the employment of MSNs was considered and it allowed relatively high processing performance to cover the disconnected nodes. This paper only presents a centralized algorithm with Lagrangean relaxation as its solution tool. Centralized algorithm, however, is not an effective method for rapid recovery of the network. For immediate and efficient recovery, we introduced an instant strategy and a centralized strategy at the same time. Our recovery strategy is more useful and realistic.

3. IP Formulation for Overlay Multicast Tree

Overlay multicast tree can be evaluated with the quality of data delivery path, the robustness of overlay multicast and control overhead at members [13]. The quality of the tree is measured with metrics such as node stress and node degrees. Since end nodes are potentially less stable than routers, for robust overlay multicast tree, it is important to alleviate the effect of member failures. Rapid recovery of the tree and packet delivery to other members are necessary. Finally, for scalability of the overlay multicast to large member groups control overhead at members should be low.

In this paper, we are interested in the robustness of overlay multicast. For the robustness two critical points are considered in the process of recovery. We want to provide multicast service without any stoppage. Therefore, the tree needs to be rapidly recovered when failure occurs. Moreover, we want multicast services with minimum delay. To satisfy these two conditions, instant and centralized recovery strategy of overlay multicast tree is introduced.

Instant strategy is an immediate solution procedure, which rapidly recovers the tree for continuous services. This strategy can be applied whenever a node leaves or joins the multicast group and causes failure in multicast tree. The strategy suggests that disconnected nodes should be attached to end nodes where delay would be the least. However, instant strategy does not guarantee overlay multicast tree with the minimum delay.

To obtain better overlay multicast tree with minimum delay centralized strategy is considered. The strategy periodically reconstructs the overlay multicast tree. Specifically, using the centralized strategy, we want to find an overlay multicast tree that minimizes maximum overlay delay for all multicast members. To guarantee the quality of data delivery path packet processing ability of each member node is restricted by degree bound. Therefore, this problem is formulated into degree-bounded minimum delay spanning tree.

Consider a graph $G=(V,E)$ where V is the set of multicast nodes and a source and E is the set of links as a full-meshed overlay network. A node $m \in V$ represents a multicast member that wants to receive services from the source. If no path exists between the source and the multicast member m, the tree recovery is impossible. By assuming at least one path between the source and node m, the failed multicast tree is recovered to connect every multicast member.

Let x_{ij}^m be a binary variable to represent a link (i,j) that is employed for a path between the source and multicast member m. If link (i,j) is employed to connect the source and node m, then $x_{ij}^m = 1$. Otherwise, $x_{ij}^m = 0$. Then the following relations hold for every node including the source s.

$$\sum_{j\neq i} x_{ij}^m - \sum_{j\neq i} x_{ji}^m = 1, \qquad \text{for all } m \in V \text{ and } i = s$$

$$\sum_{j\neq i} x_{ij}^m - \sum_{j\neq i} x_{ji}^m = 0, \qquad \text{for all } m \in V \text{ and } i \in V/\{s,m\}$$

$$\sum_{j\neq i} x_{ij}^m - \sum_{j\neq i} x_{ji}^m = -1, \qquad \text{for all } m \in V \text{ and } i = m$$

Let y_{ij} be a binary variable to represent the adoption of link (i,j) for paths in the overlay multicast tree, then the following equation holds.

$$x_{ij}^m \le y_{ij}, \quad \text{for all } m \in V \text{ and } i \neq j$$

The above constraint represents that link (i,j) has to be selected in a tree to create the path between the source and multicast member m.

Now, we consider the degree constraint of a multicast member node. By letting w_i be the degree constraint of node i, we have

$$\sum_{j\neq i} y_{ij} + \sum_{j\neq i} y_{ji} \le w_i, \qquad \text{for all } i \in V$$

To build up a spanning tree, we have

$$\sum_{j\neq i} y_{ij} = n-1, \qquad \text{for all } i \in V$$

Our objective in the multicast tree recovery is to minimize maximum overlay delay for all end hosts. By letting d_{ij} be the link delay and Z be the maximum end-to-end delay, we have the following equation to minimize Z.

$$\sum_{(i,j)\in E} d_{ij} x_{ij}^m \le Z, \qquad \text{for all } m \in V$$

From the above discussion, we have the following binary integer programming formulation.

$$\begin{aligned}
&\textit{Minimize} \quad Z \\
&\textit{subject to} \\
&\sum_{(i,j)\in E} d_{ij} x_{ij}^m \le Z && \text{for all } m \in V \\
&\sum_{j\neq i} x_{ij}^m - \sum_{j\neq i} x_{ji}^m = 1 && \text{for all } m \in V \text{ and } i = s \\
&\sum_{j\neq i} x_{ij}^m - \sum_{j\neq i} x_{ji}^m = 0 && \text{for all } m \in V \text{ and } i \in V/\{s,m\} \\
&\sum_{j\neq i} x_{ij}^m - \sum_{j\neq i} x_{ji}^m = -1 && \text{for all } m \in V \text{ and } i = m \\
&x_{ij}^m \le y_{ij} && \text{for all } m \in V \text{ and } i \neq j \\
&\sum_{j\neq i} y_{ij} + \sum_{j\neq i} y_{ji} \le w_i && \text{for all } i \in V
\end{aligned}$$

$$\sum_{j \neq i} y_{ij} = n-1 \qquad for\ all\ i \in V$$

$$x_{ij}^{m}, y_{ij} \in \{0,1\}$$

The above overlay multicast tree construction is a well-known degree bounded minimum spanning tree problem, which is NP-hard [9, 10]. Considering the NP-hardness of the overlay multicast tree problem, the proposed linear integer programming may not be effectively solved by any conventional optimization techniques. We consider a tabu search as a promising solution procedure for the above tree recovery problem. Tabu search is a powerful search heuristic that has been successfully applied to numerous combinatorial optimization problems [1]. At each step of the search, neighborhood of the current solution is explored and the best one is selected as a new solution. The search procedure does not stop even when no improvement is obtained. The best solution in the neighborhood is selected, even if it is worse than the current solution. This strategy allows the search to avoid local optima and to explore a larger fraction of the solution space.

4. Tabu search

Tabu search has achieved widespread successes in solving practical optimization problems. Applications are rapidly growing in areas such as resource management, process design, logistics, technology planning and general combinatorial optimization [2]. A tabu search incorporates three general components [1]. 1) Short-term and long-term memory structure, 2) Tabu restrictions and aspiration criteria, and 3) Intensification and diversification strategies.

Intensification strategies utilize short-term memory function to integrate features or environments of good solutions as a basis for generating still better solutions. Such strategies focus on aggressively searching for a best solution within a strategically restricted region. Diversification strategies, which typically employ a long-term memory function, redirect the search to unvisited regions of the solution space.

4.1 Initial Multicast Tree

For the overlay multicast services, the member nodes organize themselves into an initial data delivery tree. We suggest two algorithms for initialization: greedy algorithm and path attachment algorithm. We assume that the node-to-node delay is known by the source node. Each multicast member node sends hello message to the source to receive packets. This hello message includes information about node degree and delay. Thus, the source has the knowledge about all the member nodes.

In greedy algorithm the procedure starts by selecting a node with least delay from the source node. The degree of source is reduced by one. From either the source or any of the connected nodes, the procedure selects a node that can be connected with the least link delay. If the degree of any of the connected nodes reaches zero, that node is not considered in the process. This process is continued until all the nodes are connected.

Path attachment algorithm applies the degree bound after the shortest path to each node is obtained. The algorithm first finds shortest path from source to each node by Dijkstra algorithm [19]. The procedure selects the longest path and attaches it to the source. Then the procedure selects the second longest path and connects the path to the node that satisfies the degree bound with minimum delay. This process is continued until all nodes are connected. Figure 3 shows initial trees by two algorithms with overlay multicast network given in (a).

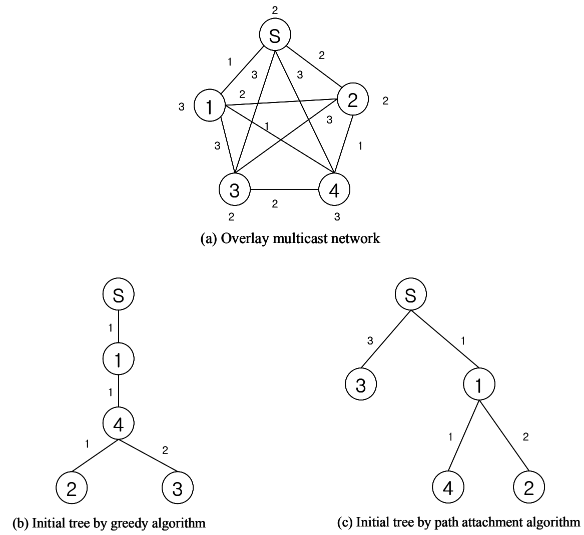

Figure 3. Example of initial trees

4.2 Intensification with short-term memory

Intensification process utilizes short-term memory to integrate features of good solutions. Such strategies focus on aggressively searching for a best solution within a strategically restricted region. A move remains tabu during a certain period (or tabu time size) to help aggressive search for better solution.

Since our objective in the dynamic overlay network is to minimize the maximum delay, at each iteration we select a node with maximum delay for a move. We consider two types of moves: “node swap” and “node reconnection”. In node reconnection a target node is reconnected to another node that satisfies the degree bound with least delay. This procedure is repeated until the solution is not improved. An end node of a path with maximum end-to-end delay is selected as the target node. When this target node does not improve the solution, the parent of the end node is considered as another target node. The parent and all its children nodes are moved in the reconnection. Note that the link delay from the reconnected node to the parent node can be improved even though the delay from the parent to its children is not changing. Nevertheless, if the solution is not improved, the procedure reconnects the end node to its grand parent node, and parent node to other node that satisfies the degree bound with least delay. Solution will be updated when it is improved. Otherwise, reconnection procedure is terminated. After the reconnection, node swap move is started. Node swap is implemented by swapping the target node with a node with least delay. This swap move thus allows on uphill move in the tabu search for later improvement. After this swap move, reconnection and swap is repeated.

4.3 Diversification with long-term memory

Diversification strategies, as their name suggests, are designed to drive the search into new regions. In general, the performance of tabu search becomes significantly stronger by including long term memory and its associated strategies. In this study frequency-based memory is used for diversification. Links that are not frequently in the previous multicast trees are chosen for a new tree. To restart the search after each pass, the procedure excludes n-1 most frequently used links during the pass. Rest of the links is considered to have an initial tree for diversification.

4.4 Overall tabu search procedure

Based on the intensification and diversification strategies in the previous sections, we describe the overall procedure of the tabu search. First, we generate initial overlay multicast tree. Starting from the initial tree move operations are implemented. Whenever a move is implemented, we update the current solution *S_current* and the best solution *S_best*. Tabu list and the frequency of each link in the solution $F(i, j)$ are also updated. The intensification process is continued as far as the best solution is improved. When moves result in no improvement consecutively for *N_max* iterations the diversification process is performed with a new overlay multicast tree. When the number of diversification reaches *D_max*, the overall procedure is terminated. The overall tabu search procedure is explained as follows.

Step1 Initial tree generation
Obtain initial overlay multicast tree by algorithms in Section 4.1
Step2 Initial parameters of tabu search
S_current and *S_best* is initialized as 0
Set *N*=0, *D*=0 and $F(i, j) = 0$
Tabu list is empty
Step3 Intensification
Select a target to apply move
Apply reconnection moves until the solution is not improved
Apply swap move
Update *S_current*
Update *S_best*, tabu list and $F(i, j)$
If $N < N\_max$, repeat this step
If $N > N\_max$ and $D < D\_max$, go to step4
If $D > D\_max$, stop the tabu search
Step4 Diversification
Obtain a new tree using $F(i, j)$
Restart the tabu search with the initial tree
Set *N=0* and *F (i, j)* =0
Go to Step3

5. Computational Results

In this section, we discuss the computational results of the tabu search for the degree-bounded minimum delay spanning tree. Four different sizes of overlay multicast networks are generated each with 10, 30, 50 and 100 nodes. The node degree is assumed to be ranged 2 ~ 4. In each network, links are randomly generated such that the delay of each overlay link is ranged 1 ~ 10 by assuming maximum 10 physical hops. The algorithm for Tabu search is implemented in C language and run on a 2.40 GHz Intel Pentium 4 based on personal computer with 512 Mbyte of memory under Windows XP.

We first test two initial solution strategies: greedy algorithm and path attachment algorithm. Ten problems with 50 nodes are tested as in Table 1. As shown in the table, path attachment algorithm is superior to the greedy algorithm. We thus start the tabu search with the initial solution by the path attachment algorithm.

Before testing the performance of our tabu search, we need to tune the tabu parameters: tabu list size, *N_max* for the intensification procedure and *D_max* of the diversification. Tabu list size represents the number of iterations during which a target node is forbidden to be adopted in move operation. By assuming that an appropriate tabu list size is proportional to the number of nodes, we perform tests with 50 nodes. Figure 4 shows that 0.2N is suitable for tabu list size with 50 nodes.

Test for N_max is performed as in Figure 5. The figure shows that appropriate value for N_max is 0.6N for 10 nodes, 0.6N for 30 nodes, 0.8N for 50 nodes and 0.8N for 100 nodes. The test on D_max is also performed as in Figure 6. By testing ten problems the portion that gibes no further improvement for the successive diversification is plotted in the figure. From the figure, it seems to be reasonable to perform two diversifications for problems with 10 nodes, three for 30 nodes, six for 50 nodes and nine for 100 nodes.

Computational results of the proposed tabu search are shown in Table 2,3,4,5. CPLEX [14] is employed to compare the solutions. Due to the exponential growth of the branches in the process of CPLEX, it fails to obtain the optimal solution even with 10,000 seconds running time for networks with 30, 50 and 100 nodes. Especially, in the network with 100 nodes CPLEX experiences memory problem. Table 2 shows the results of tabu search with 10 nodes. The proposed tabu search generated optimal solutions in all cases with 10 nodes. Table 3 and 4 show the results of tabu search with 30 and 50 nodes respectively. The proposed tabu search generated optimal solutions in two cases with 30 and 50 nodes. The gap in worst case reaches 15.4% in problems with 30 and 50 nodes. Table 5 shows the results of tabu search with 100 nodes. CPLEX is out of memory in the problems with 100 nodes. Thus, we fail to obtain the near optimal solution. Computational experiments are performed in overlay multicast network 10, 30, 50 and 100 nodes. The efficiency of tabu search in CPU seconds is illustrated as the number of multicast nodes increases.

The effectiveness of diversification in this tabu search is demonstrated in Figure 7. The figure shows results of tabu search with and without diversification. With diversification, the maximum end-to-end delay of the overlay multicast tree is decreased approximately by 1.5% for 10 nodes, 2.9% for 30 nodes, 3.7% for 50 nodes and 3.5% for 100 nodes.

We have additionally carried out tests in order to observe the effect of node degree bound. Figure 8 shows the effect of no degree. To analyze the effect of node degree, the degree of all nodes is fixed to 2, 3 and 4 respectively except for the source node. As shown in figure 8, the decrease of average maximum delay seems to be sensitive to the node degree in problems with 10, 30, 50 and 100 nodes.

Table 1. Test of Feasible Initial Solutions

Problem	Greedy	Path attachment
1	31	31
2	29	26
3	30	28
4	30	32
5	24	24
6	23	24
7	36	33
8	29	27
9	30	25
10	29	27
Average	29.1	27.7

Table 2. Computational results of tabu search with 10 nodes

Problem	Tabu Search Solution (TS)	CPU seconds	CPLEX Solution (OPT)	CPU seconds	GAP*
1	8	1.11	8	5.55	0.000
2	6	1.01	6	0.34	0.000
3	6	1.12	6	0.17	0.000
4	8	1.10	8	0.27	0.000
5	10	1.13	10	12.38	0.000
6	9	1.12	9	0.2	0.000
7	9	1.03	9	7.56	0.000
8	6	1.05	6	0.34	0.000
9	7	1.12	7	0.16	0.000
10	7	1.09	7	0.17	0.000

* GAP: (TS-OPT)/OPT

Table 3. Computational results of tabu search with 30 nodes

Problem	Tabu Search Solution (TS)	CPU seconds	CPLEX Solution (OPT)	CPU seconds	GAP**
1	16	60.08	15	10,000*	0.067
2	18	62.94	17	10,000*	0.059
3	17	60.39	17	10,000*	0.000
4	15	60.56	13	10,000*	0.154
5	16	60.25	15	10,000*	0.067
6	17	62.78	17	10,000*	0.000
7	17	63.23	16	10,000*	0.063
8	18	61.86	17	10,000*	0.059
9	16	62.00	15	10,000*	0.067
10	18	60.39	16	10,000*	0.125

*: Terminated by the time limit, ** GAP: (TS-OPT)/OPT

Table 4. Computational results of tabu search with 50 nodes

Problem	Tabu Search Solution (TS)	CPU seconds	CPLEX Solution (OPT)	CPU seconds	GAP**
1	29	73.76	27	10,000*	0.074
2	24	73.44	24	10,000*	0.000
3	26	74.19	23	10,000*	0.130
4	29	74.12	28	10,000*	0.036
5	23	74.60	22	10,000*	0.045
6	22	73.48	20	10,000*	0.100
7	30	74.05	26	10,000*	0.154
8	24	73.01	22	10,000*	0.091
9	24	73.36	22	10,000*	0.091
10	24	74.09	24	10,000*	0.000

*: Terminated by the time limit, ** GAP: (TS-OPT)/OPT

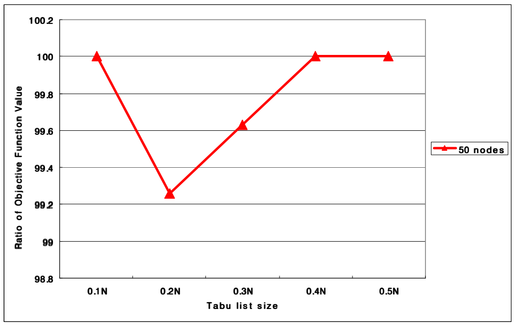

Figure 4. Test of tabu list size with 50 nodes

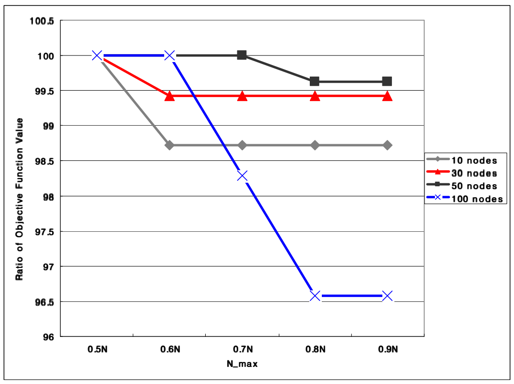

Figure 5. Test of N_max

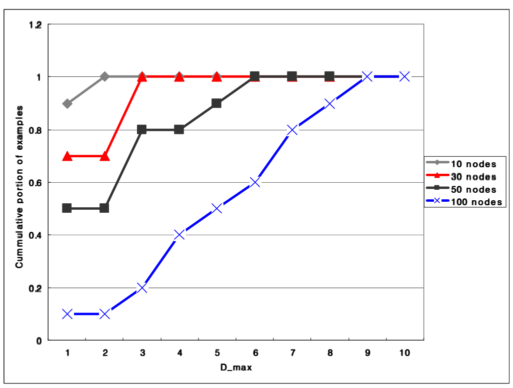

Figure 6. Test of D_max

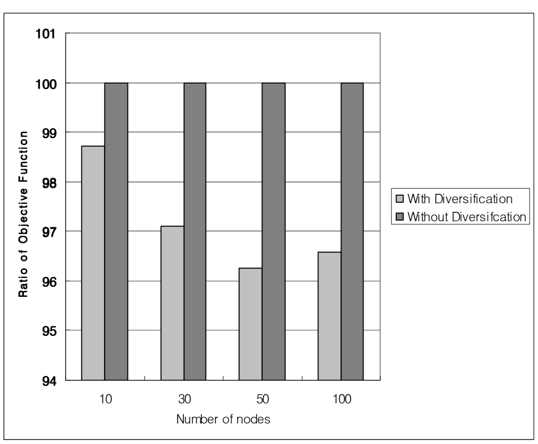

Figure 7. Effectiveness of Diversification with 10, 30, 50 and 100 nodes

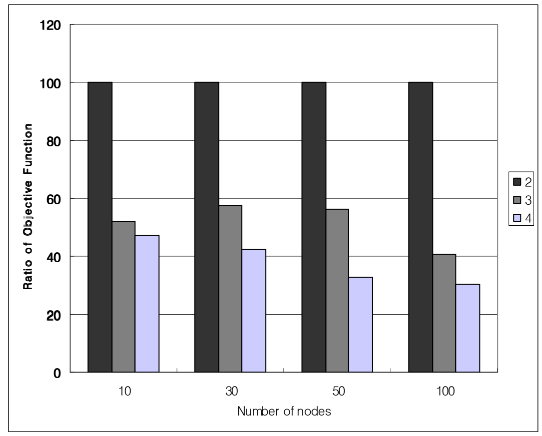

Figure 8. Effect of degree with 10, 30, 50 and 100 nodes

6. Conclusion

We have presented two efficient recovery strategies of multicast tree in the end-to-end overlay network. An instant strategy was introduced to provide rapid recovery and a centralized strategy was proposed to achieve performance close to the optimal level. Overlay multicast tree was represented as a degree-bounded minimum maximum-delay spanning tree, which is a problem known as NP-hard. Thus, a tabu search was introduced as part of the centralized strategy. We compared the results of the suggested tabu search with that of the IP formulation with several simulation studies. As the result, the proposed tabu search and the strategy as a whole showed excellent results, where the solution was very close to the optimal, but derived much more rapidly.

References

[1] F. Glover, “ Tabu search,” ORSA Journal on Computing, Vol. 1, No. 3, pp.190-206, Summer, 1989.
[2] E. Rolland, H. Pirkul, and F. Glover, “Tabu search for graph partitioning,” Annals of Operations Research, Vol. 63, pp. 209-232, 1996.
[3] P. Francis, “Yoid: Your own Internet distribution,”, UC Berkeley ACIRI Tech Report, Apr., 2000.
[4] Y. Chawathe, “Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service”, Ph. D. thesis, Department of EECS, UC Berkeley, Dec., 2000.
[5] Y. Chu, S. Rao, and H. Zhang, “A case for end system multicast”, ACM Sigmetrics, 2000.
[6] J. Jannotti, D. K. Gifford, K. Jhonson, and M. Kaashock, “Overcast: Reliable multicasting with an overlay network,” 5th Symposium on Opening System Design and Implementation, Dec., 2000.
[7] D. Helder and S. Jamin, “Banana Tree Protocol, an End-host Multicast Protocol”, Technical Report, July 2000.
[8] D. Tellium, “ALMI: An Application Level Multicast Infrastructure”, 3rd Usenix Symposium on Internet Technologies and Systems, March, 2001.
[9] S, Shi, J. Turner, and M. Waldvogel, “Dimensioning server access bandwidth and multicast and multicast routing in overlay networks”, in Proceedings of NOSSDAV, June, 2001.
[10] E. Nardelli and G. Proietti, “Finding all the best swaps of a minimum diameter spanning tree under transient edge

failures", Journal of Graph Algorithms and Applications, Vol. 5, No. 5, pp. 39-57, 2001.

[11] N.M. Malouch, Z. Li, D. Rubenstein and S.A Sahu, "Graph theoretic approach in proxy-assisted, end-system multicast", Quality of Service, 2002. Tenth IEEE International Workshop, pp. 106-115, May 2002.

[12] S. Shi and J. Turner, "Routing in Overlay Multicast Networks", IEEE INFOCOM, 2002.

[13] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast", in Proc. of ACM sigcomm, Aug., 2002.

[14] CPLEX 8.1, CPLEX Optimization Inc., 2003.

[15] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee and S. Khuller, "Construction of an efficient overlay multicast infrastructure for real-time applications", IEEE INFOCOM, 2003.

[16] S.Fahmy and Min S. Kwon, "Characterizing overlay multicast networks", In Proceedings of 11th IEEE International Conference on Network Protocols, 2003.

[17] Hee K. Cho and Chae Y. Lee, "Multicast Tree Rearrangement to Recover Node Failure in Overlay Multicast Network", Computers and Operations Research, 2004.

[18] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks", IEEE Journal on Selected Areas in Communications, Vol. 22, No. 1, January, 2004.

[19] J. A. Bondy and U. S. R. Murty, Graph Theory with Applications, North-Holland, 1976.