韓國經營科學會誌
第33권 第2號
2008年 6月

실시간 인터넷 서비스를 위한 오브레이 말티케스트 트리의 패스 신뢰성 최대화

이정훈* · 이채영**†

Maximization of Path Reliabilities in Overlay Multicast Trees for Realtime Internet Service

Jung H. Lee* · Chae Y. Lee**

■ Abstract ■

Overlay Multicast is a promising approach to overcome the implementation problem of IP multicast. Real time services like Internet broadcasting are provided by the overlay multicast technology due to the complex nature and high cost of IP multicast. To reduce frequent updates of multicast members and to support real time service without delay, we suggest a reliable overlay multicast tree based on members' sojourn probabilities. Path reliabilities from a source to member nodes are considered to maximize the reliability of an overlay multicast tree. The problem is formulated as a binary integer programming with degree and delay bounds. A tabu search heuristic is developed to solve the NP-complete problem. Outstanding results are obtained which is comparable to the optimal solution and applicable in real time.

Keyword : Overlay Multicast, Multicast Tree, Path Reliability, Tabu Search

논문접수일 : 2007년 11월 08일 논문게재확정일 : 2008년 05월 19일
논문수정일(1차 : 2008년 04월 28일)
* ㈜한국기업평가
** 한국과학기술원 산업공학과
† 교신저자

1. Introduction

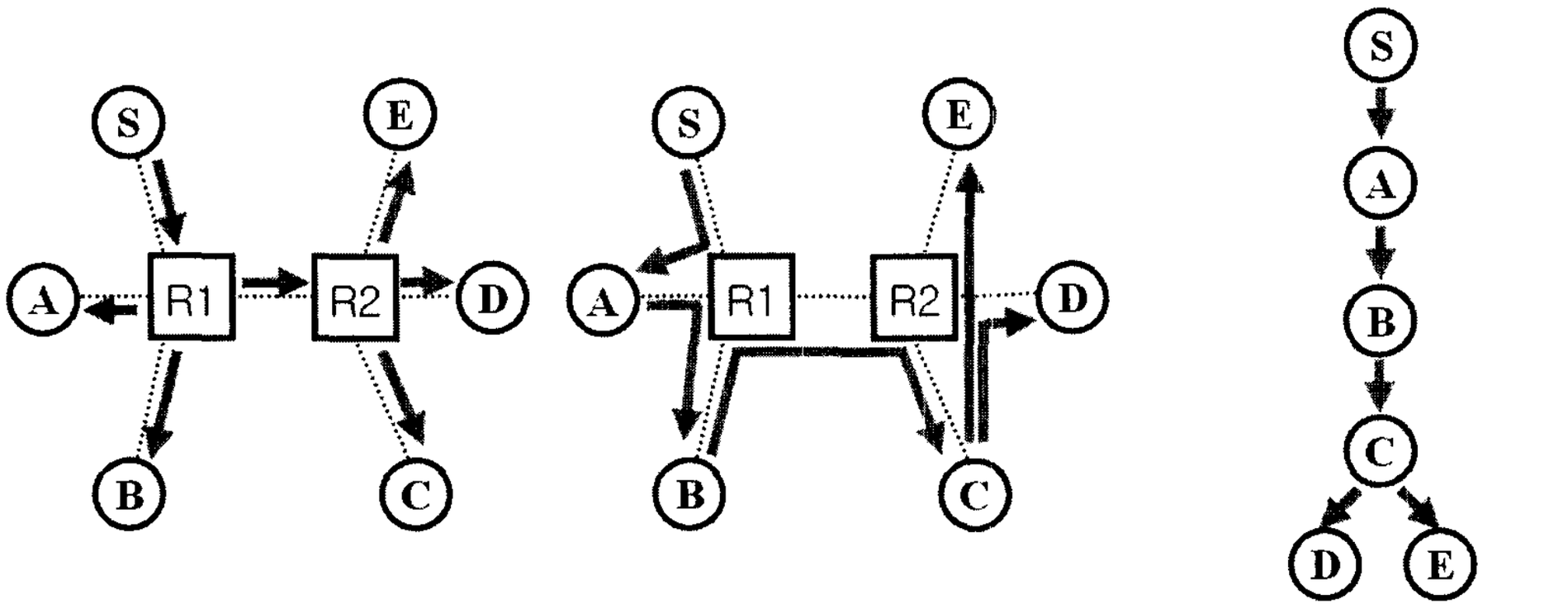
Multicasting in telecommunication has become a critical issue of many next generation applications, including video-on-demand (VOD) and IPTV. Many emerging applications on the Internet are characterized by high volume data rate and multiple receivers. Multicast is an efficient method to transmit the same information to a group of receivers simultaneously [1].

In traditional IP multicast, routers played an important role in replication and transmission of data packets to receivers. Basic concept of IP multicast is presented in [Figure 1 (a)]. Source node S sends packets to its adjacent router R1 or R2. Each router then copies received packets and transmits them to receivers or other routers. This IP multicast is efficient in bandwidth consumption. However, most commercial ISPs have not deployed routers to support IP multicast due to the high implementing cost and complexity of the technology [2, 3].

Overlay multicast is widely examined as an alternative to the IP multicast. Differently from the IP multicast, overlay multicast can be implemented in application layer [1]. After a source node transmits packets to an end node, the receiving end node replicates and sends packets to other members. This is illustrated in [Figure 1 (b)]. In the application layer, a logical overlay multicast tree structure is built as in [Figure 1 (c)].

There are two major research issues in the overlay multicast. One is to minimize the delay from a multicast source to other members. Lee, Park and Baek [4] studied minimizing maximum delay in dynamic overlay network. The problem is formulated as a degree-bounded minimum spanning tree. A tabu search heuristic is developed. Shi and Turner [5] proposed a heuristic multicast routing algorithm for overlay networks which optimizes the access bandwidth usage, while satisfying the end-to-end delay requirements of applications.

Another issue is related to the reliability of overlay multicast trees. Cho and Lee [6] examined multicast tree rearrangement to recover node failures. Lee and Kim [7] designed reliable overlay multicast trees with multiple sessions. The objective is to build a reliable multicast tree for each session that satisfies common con-



(a) IP Multicast (b) Overlay Multicast (c) Overlay Multicast tree

[Figure 1] IP and Overlay Multicast

strains of an overlay network. Link level reliability is considered in building overlay multicast trees in which the minimum link reliability is maximized.

In this paper, we are interested in path reliability in building overlay multicast trees. Since an overlay multicast tree can be disconnected due to the leave of multicast members, the end to end path level reliability is important to deliver data packets without failure. Sojourn probability of each member node in a service session is considered to build reliable paths from the source to members. The overlay multicast tree needs to be updated periodically to solve the disconnection problem and to increase the path reliabilities.

The rest of this paper is organized as follows. In Section 2, we formulate the construction of overlay multicast tree that considers path level reliability. In Section 3, a tabu search based algorithm is developed to solve the NP-hard problem. Computational results and conclusion are provided in Section 4 and 5 respectively.

2. Path Reliabilities in Overlay Multicast Tree

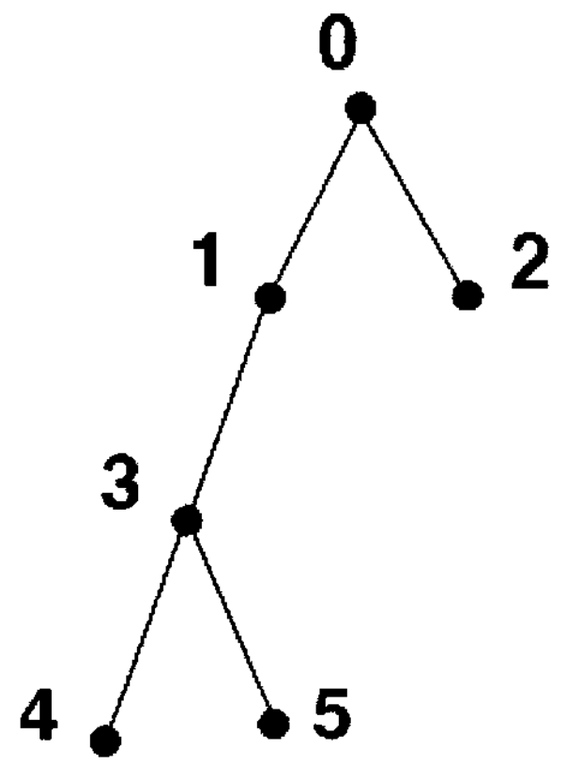
When a member node joins or leaves a multicast group, it needs to be connected into or disconnected from its multicast tree. Joining of a new member can be handled by connecting it to an existing member with enough node capacity. However, updating procedure is not simple when a member node leaves a multicast group. If a member node leaves a multicast group, all descendent nodes of the member are disconnected. Therefore, we need to consider reliability to build a sustainable overlay multicast tree with reduced updates.

Link reliability is considered in [7] to increase reliability of a multicast tree. Sojourn time of two member nodes is considered to have the link reliability. Each node has a sojourn probability in a session of Internet service that resides in its multicast group during a fixed period of time. A tabu search heuristic is employed to maximize the minimum link reliability. However, an end-to-end reliability from a source to a member node is more dependent on the path reliability than on the link reliability in the overlay multicast.

Given a multicast tree, the path reliability from a source to a member node can be obtained by multiplying node reliabilities in the path. Now, to have a sustainable multicast tree even with the departure of a member, it is important to connect nodes with high sojourn probabilities near to the source. For that purpose the reliability of the multicast tree is obtained by the combination of all path reliabilities. As an example the reliability of an overlay multicast tree given in [Figure 2] with the five paths is computed as $p_1^4 * p_2 * p_3^3 * p_4 * p_5$, where p_j is the sojourn probability of node j. The probability of source node is assumed to be one. It is clear from the example that the reliability is highly dependent on the nodes directly connected to the source and those in the upper level of the tree. In order to have the sojourn probability it is essential to operate a historical log file [7, 14, 15] which includes information of log-in, log-out, and service rate of each member node.

An overlay multicast tree can be modeled with a graph $G = (V, E)$, where V represents a set of multicast member nodes and E represents links among nodes in the multicast network. To service a member node $m_i \in V$ in the network a path is necessary to connect it to the source node s.

Let x_{ij} be a binary variable for link (i, j). If there is a link between node i and j in the multicast tree, $x_{ij} = 1$. Otherwise, $x_{ij} = 0$. Also, let y_{ijm} be a binary variable to represent a path between the source and multicast member m. $y_{ijm} = 1$, if there is a direct link from node i to j on the path fro source s to node m. Otherwise, $y_{ijm} = 0$. Then the following flow conservation equations hold for every node in multicast network.



[Figure 2] An example overlay multicast tree

$$\sum_{j \neq i} y_{ijm} - \sum_{j \neq i} y_{jim} = \begin{cases} +1, & \text{if } i = s, \text{ for } i, j, m \in V \\ -1, & \text{if } i = m, \text{ for } i, j, m \in V \\ 0, & otherwise \end{cases}$$

For a link (i, j) to be inserted in the path from the source to node m, the link has to be selected for the multicast tree as in the constraint below.

$$y_{ijm} \leq x_{ij}, \quad \text{for } m \in V \text{ and } (i, j) \in E$$

Since a multicast tree is a spanning tree with n members, we have

$$\sum_{(i, j) \in E} x_{ij} = n - 1,$$

To satisfy the end-to-end delay bound for each member m, the following constraint is necessary.

$$\sum_{(i, j) \in E} d_{ij} y_{ijm} \leq L, \qquad \text{for } m \in V$$

where d_{ij} is the delay in link (i, j) and L is the delay bound. If $d_{ij} = 1$ for all link, L becomes the end-to-end hop counts in multicast tree.

Now, each member node in the overlay network has capacity limit which can be represented by a degree constraint. Let D_i be the degree constraint of node i, then we have

$$\sum_{j \neq i} x_{ij} + \sum_{j \neq i} x_{ji} \leq D_i, \qquad \text{for } i, j \in V$$

In an overlay multicast tree, service data rate is restricted by the link capacity. Let r be the service data rate and C_{ij} be the capacity of link (i, j). Then the link capacity constraint becomes

$$r x_{ij} \leq C_{ij}, \qquad \text{for } (i, j) \in E$$

Now, our objective is to have an overlay multicast tree that maximizes the path reliabilities. Since we have multiple paths to connect members in the multicast tree, it is reasonable to maximize the reliability of all paths. Note that the path reliability is the multiplication of the node sojourn probability p_j in the path. Thus, we are interested in maximizing the reliability of the overlay multicast tree given by $\prod_j p_j^{\sum_{m \in V} \sum_{(i, j) \in E} y_{ijm}}$. In other words, the multicast tree reliability is computed with the probability p_j of each node j and its frequency $\sum_{m \in V} \sum_{(i, j) \in E} y_{ijm}$ in the paths from the

source to member nodes. Here, we employ a logarithmic function of the reliability which is usually adopted to represent various utilities. Then, our objective function becomes

$$\text{Maximiae } \sum_j \left(\sum_{m \in V} \sum_{(i,j) \in E} y_{ijm} \right) \log p_j$$

From the above discussion, we have the following binary integer programming formulation to have a reliable overlay multicast tree.

$$\text{Maximize } \sum_j \left(\sum_{m \in V} \sum_{(i,j) \in E} y_{ijm} \right) \log p_j$$

subject to

$$\sum_{j \neq i} y_{ijm} - \sum_{j \neq i} y_{jim} = \begin{cases} +1, & \text{if } i = s, \text{ for } i, j, m \in V \\ -1, & \text{if } i = m, \text{ for } i, j, m \in V \\ 0, & \textit{otherwise} \end{cases}$$

$$y_{ijm} \leq x_{ij}, \qquad \text{for m} \in V \text{ and } (i,j) \in E$$

$$\sum_{(i,j) \in E} x_{ij} = n - 1,$$

$$\sum_{(i,j) \in E} d_{ij} y_{ijm} \leq L, \qquad \text{for } m \in V$$

$$\sum_{j \neq i} x_{ij} + \sum_{j \neq i} x_{ji} \leq D_i, \qquad \text{for } i, j \in V$$

$$r\, x_{ij} \leq C_{ij}, \qquad \text{for } (i,j) \in E$$

$$x_{ij},\ y_{ijm} \in \{0, 1\}$$

Note that the model is to find the degree constrained spanning tree which maximizes the path reliabilities and that the degree constrained spanning tree is well-known NP-complete [8, 9] problem. This implies that any known algorithm will run in time exponential in the size of problem instance. Such an algorithm is thus in most cases unusable for real-world size problems. As encouraging results on NP-hard problems, we investigate a tabu search heuristic to have a reliable overlay multicast tree which reflects the path level reliability.

3. Tabu Search

Tabu search [10, 11] is a successful meta-heuristic method to solve complex optimization problems. The main idea of tabu search includes three general components. 1) Initial solution, 2) Intensification with a short-term memory, 3) Diversification with a long-term memory.

3.1 Initial Overlay Multicast tree

To build an initial overlay multicast tree, we propose the following two procedures : "Degree-first initial tree" and "Random initial tree." For the Degree-first initial tree, member nodes are sequenced in nonincreasing order of node degree. When more than two nodes have the same degree, nodes are sequenced in nonincreasing order of sojourn probability. A spanning tree is built by selecting nodes in the sequence. Each selected node is connected to a node with higher node degree and sojourn probability to increase the end-to-end reliability. A tree constructed by this procedure may not satisfy the delay bound or link capacity constraints. To have a feasible solution, a node that does not satisfy the constraints is selected. The selected node and its descendants are reconnected to an ascendant node which satisfies the constraints. For the random initial tree, a tree is built by randomly selecting nodes to connect to the source. Reconnection process continues until all nodes satisfy given constraints.

3.2 Intensification

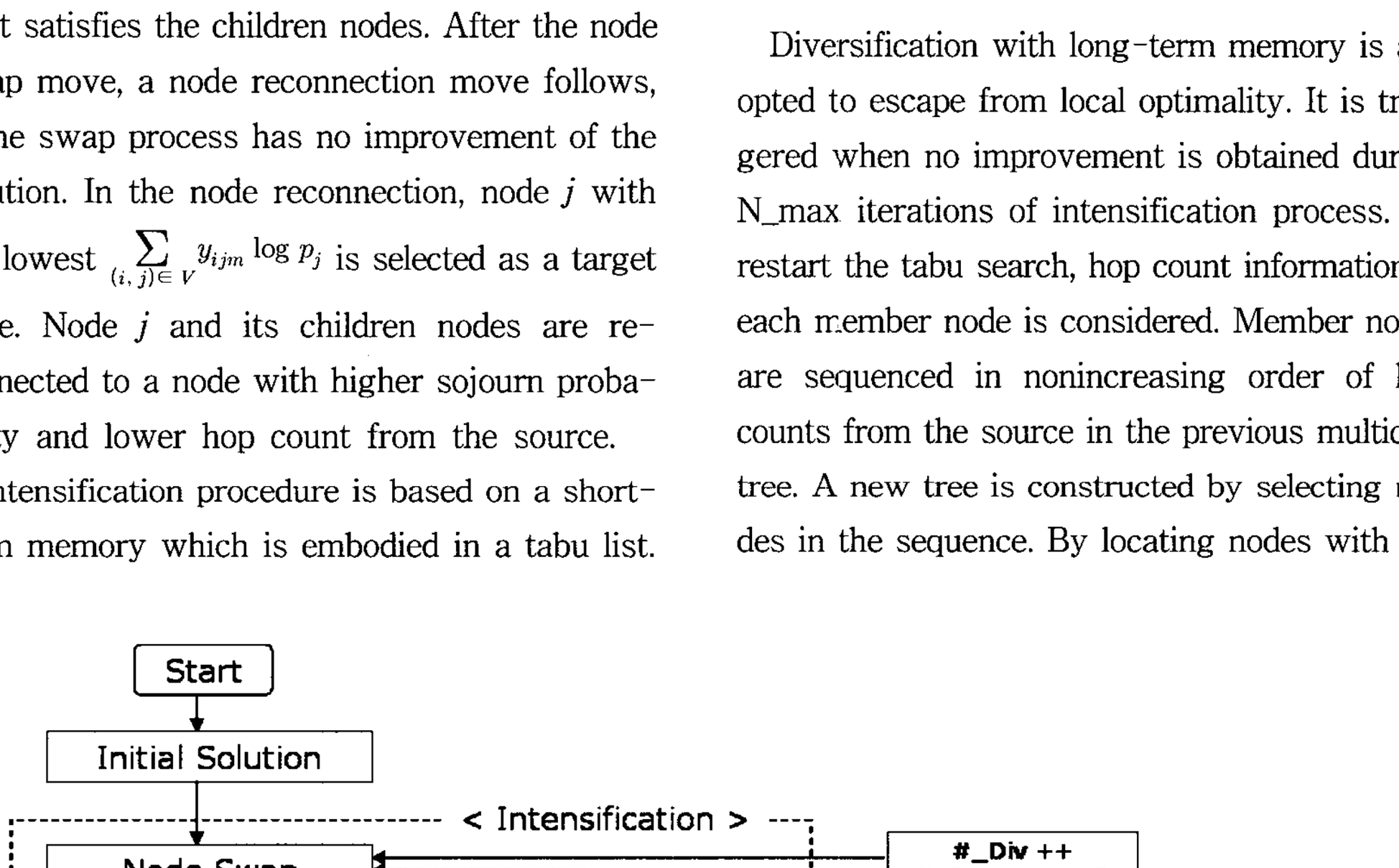
For intensification process we consider two

types of moves : "node swap" and "node reconnection." In node swap move, node j with the lowest $\sum_{m \in V} \sum_{i \in V} y_{ijm} \log p_j$ is selected as a target node. Since $\log p_j \leq 0$, a target node which has lower sojourn probability and supports many children nodes is selected for the swap move. This is because the target node lower sojourn probability of all its descendants. Target node j is exchanged with node i that has higher sojourn probability even with lower degree bound as far as it satisfies the children nodes. After the node swap move, a node reconnection move follows, if the swap process has no improvement of the solution. In the node reconnection, node j with the lowest $\sum_{(i,j) \in V} y_{ijm} \log p_j$ is selected as a target node. Node j and its children nodes are reconnected to a node with higher sojourn probability and lower hop count from the source.

Intensification procedure is based on a short-term memory which is embodied in a tabu list. After applying the node swap or node reconnection move, the corresponding target node is added to the tabu list. Nodes in tabu list are prohibited for a certain period to be selected again as a target node for the next intensification process. Intensification procedure is repeated until no solution improvement is obtained consecutively for N_max iterations as in [Figure 3].

3.3 Diversification

Diversification with long-term memory is adopted to escape from local optimality. It is triggered when no improvement is obtained during N_max iterations of intensification process. To restart the tabu search, hop count information of each member node is considered. Member nodes are sequenced in nonincreasing order of hop counts from the source in the previous multicast tree. A new tree is constructed by selecting nodes in the sequence. By locating nodes with hi-



[Figure 3] Proposed tabu search procedure

gher hop counts closer to the source node, we expect to build a new multicast tree which has not been constructed in the intensification process. The tabu search then continues with the intensification process in Section 3.2. When the number of diversifications is equal to D_max, the procedure is terminated.

4. Computational Result

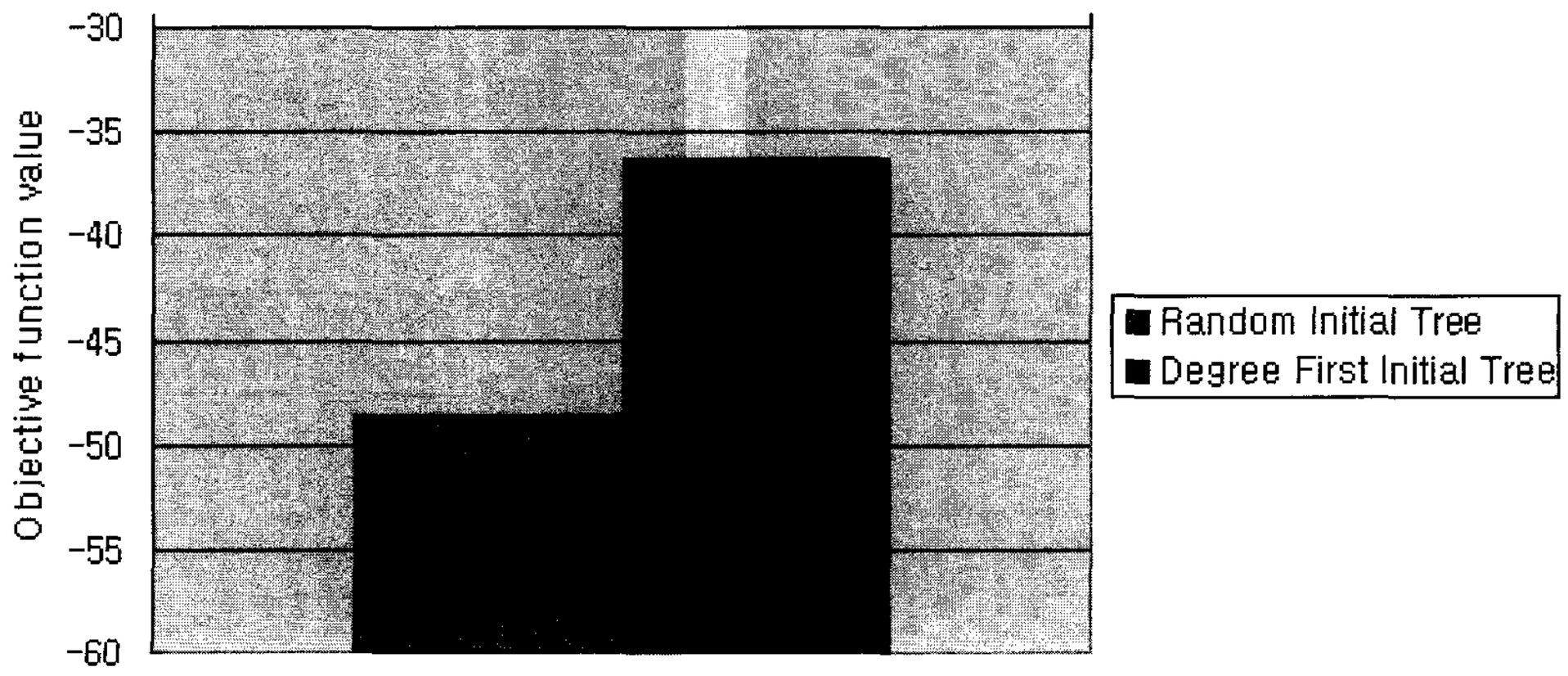
To test the proposed tabu search to have a reliable overlay multicast tree, four different size of overlay multicast networks are considered. Problems with 10, 50, 100 and 200 nodes are designed. For each case, 10 problems are generated by randomly selecting sojourn probability, node degree, link capacity and delay bound as in <Table 1> Widely spread video streaming codec MPEG-I [12] is assumed with service data rate 384 kbps or 600kbps. The sojourn probability of each member node is exponentially distributed with $p_i = 1 - e^{-\lambda_i}$, where λ_i is failure rate of a member node i. When λ_i is distributed over [0.5 ~5.0], sojourn probability p_i is in the range of [0.39~0.99]. All procedures are run on a Pentium IV-3.0 GHz PC with 1024Mbytes of memory.

〈Table 1〉 Parameters for Overlay Multicast trees

◦ Number of nodes(n) :	10, 50, 100, 200 nodes
◦ Failure rate(λ_i) :	0.5~5.0
◦ Sojourn probability(p_i) :	$p_i = 1 - e^{-\lambda_i}$
◦ Note degree(D_i) :	3, 4, 5
◦ Link delay(d_{ij}) :	1 hop
◦ Delay bound(L) :	0.3n, 0.5n
◦ Service data rate(r) :	384kbps, 600kbps
◦ Link capacity(C_{ij}) :	0.5Mbps~1Mmps

We first test out two initial tree construction strategies : Degree-first initial tree and Random initial tree. Problems with 50 nodes are tested. Average objective function value of ten problems by each strategy is shown as in [Figure 4]. The figure shows that Degree-first initial tree gives better solution than the random method. Thus, we adopt Degree-first initial tree for the rest of experiments.

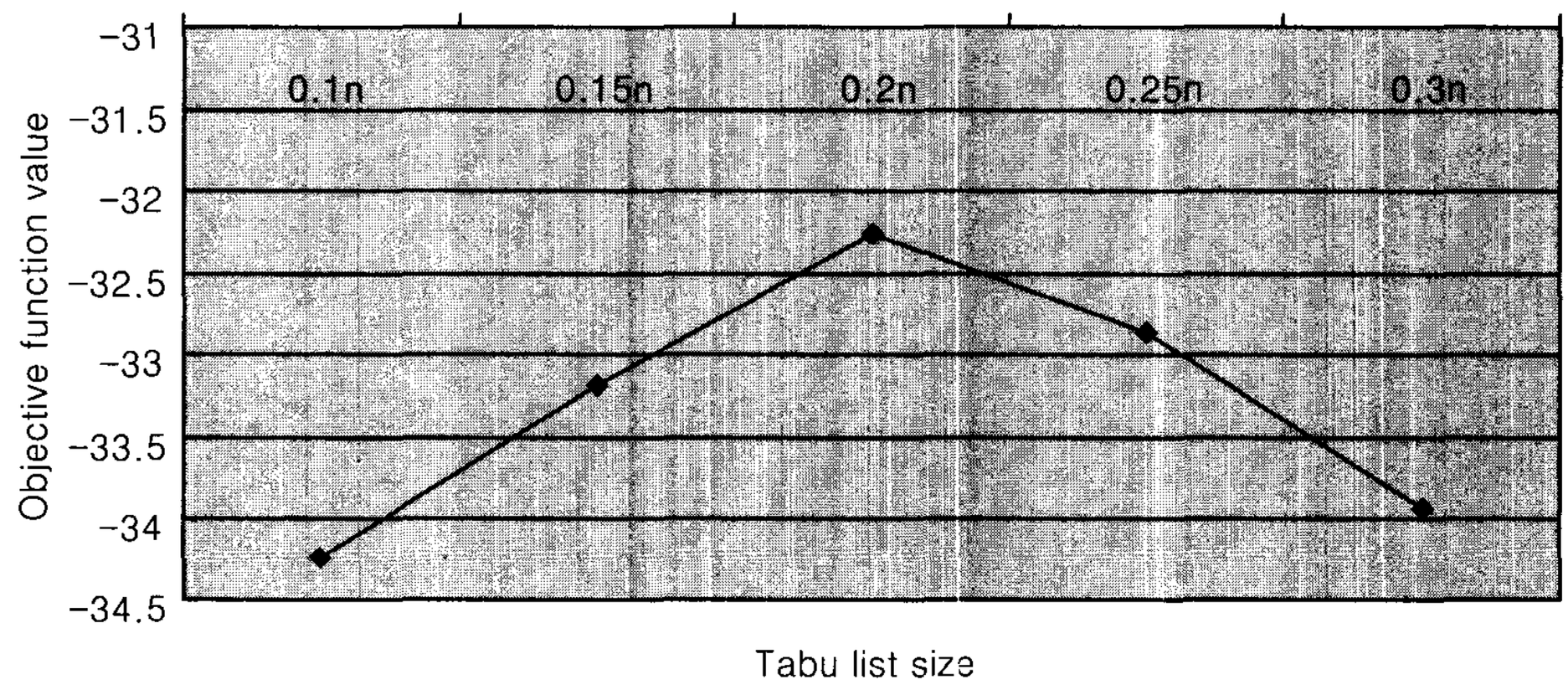
Before applying tabu search, we need to tune tabu parameters. Tabu list size, N_max for intensification and D_max for diversification are experimented with 50-node problems. A tabu list



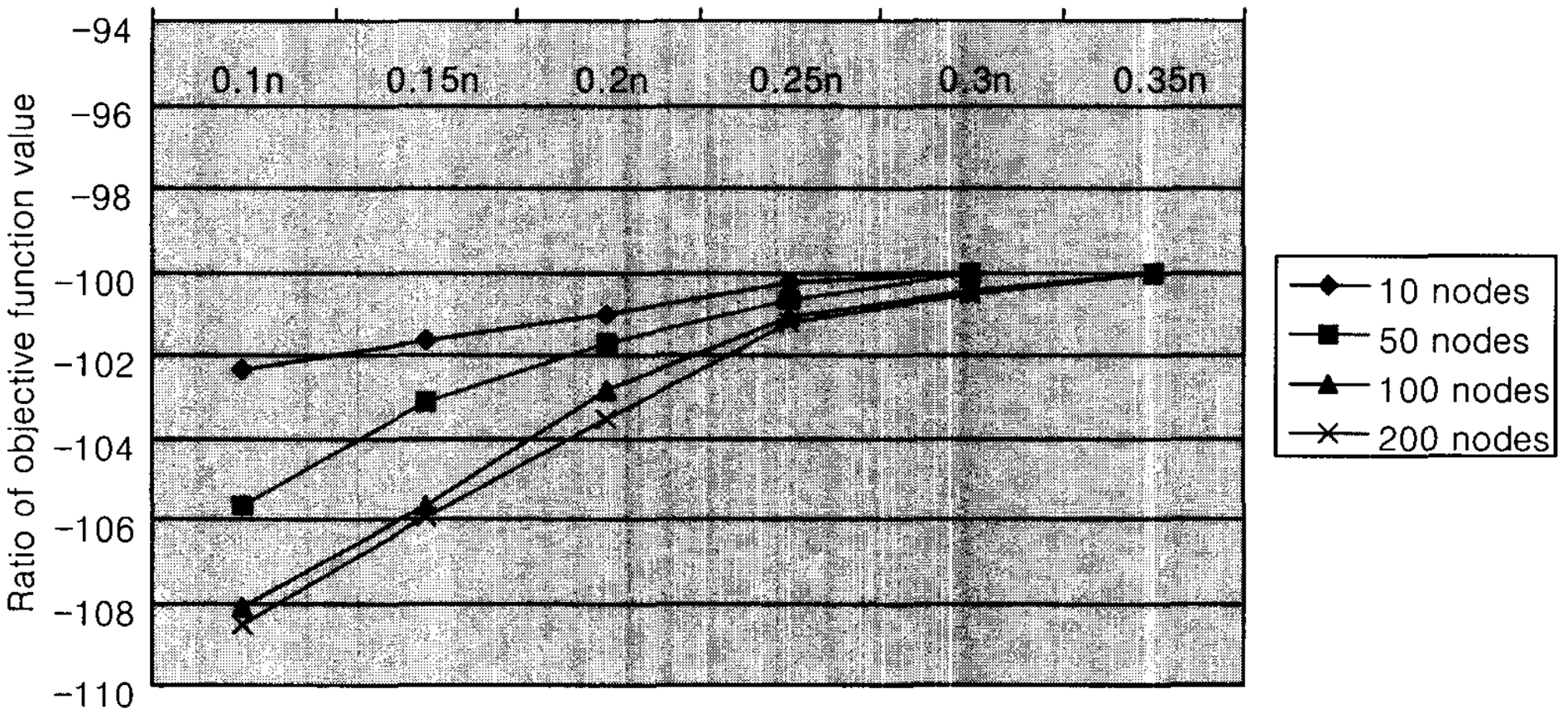
[Figure 4] Test of Initial Tree

size represents the number of iterations during which a target node selected is forbidden to be selected again. [Figure 5] shows that tabu list size of 10 gives the best result in 50-node problems. By assuming that the appropriate tabu list size is proportional to the number of nodes, tabu size of 0.2n is employed for other problems.

Test for N_max is performed as in [Figure 6]. The figure shows that appropriate value for N_max is 0.3n. The number of diversification is related to the solution quality in tabu search. Test of D_max is performed as in [Figure 7]. Among ten problems, the portion that gives no further improvement for the successive diversification is plotted in the figure. Clearly, more diversification is required as the number of nodes increases. From the experiments it seems to be reasonable to apply D_max = 4 for 10-node problem and D_max = 8 for other problems.

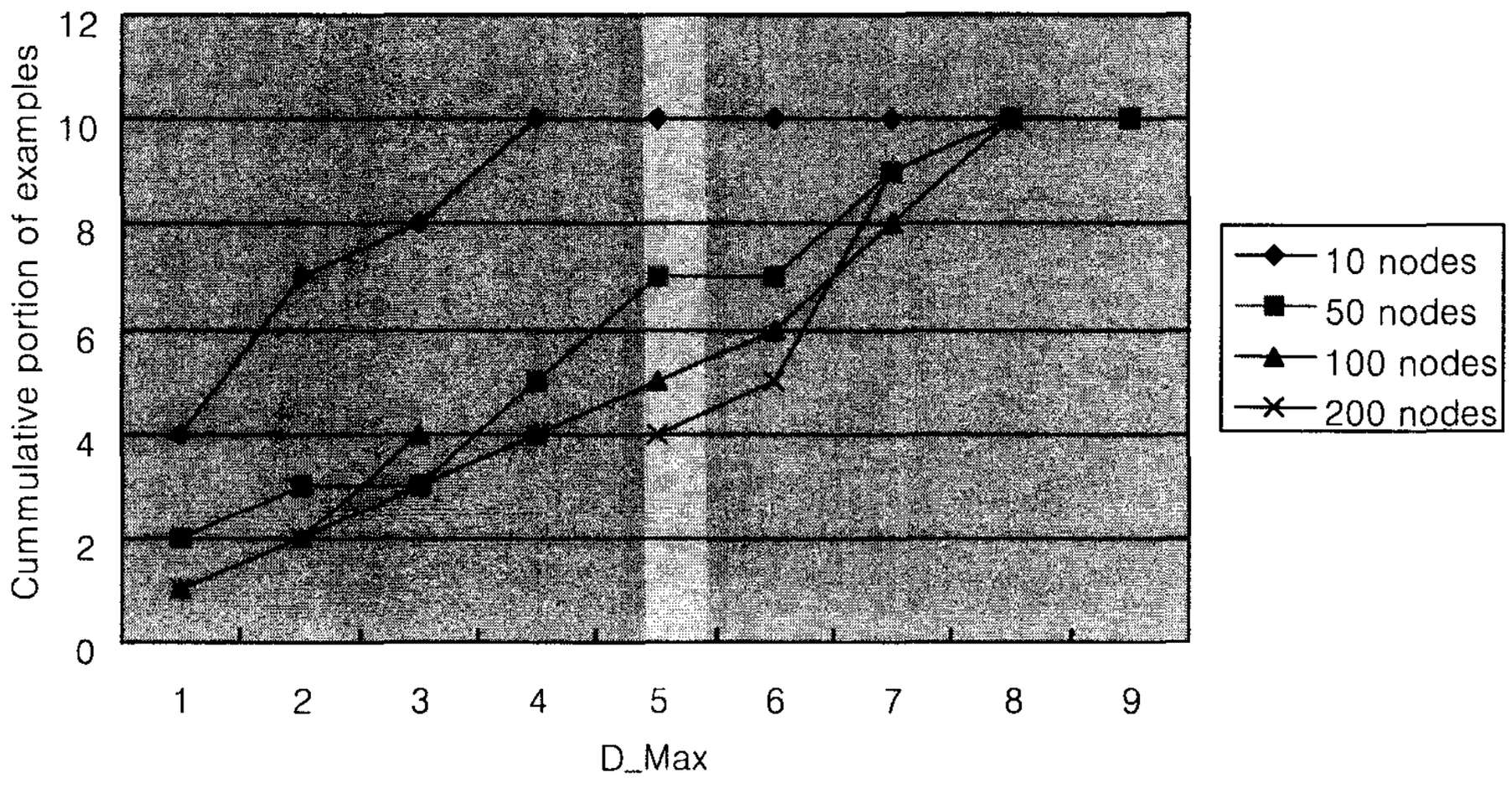


[Figure 5] Test of tabu list size

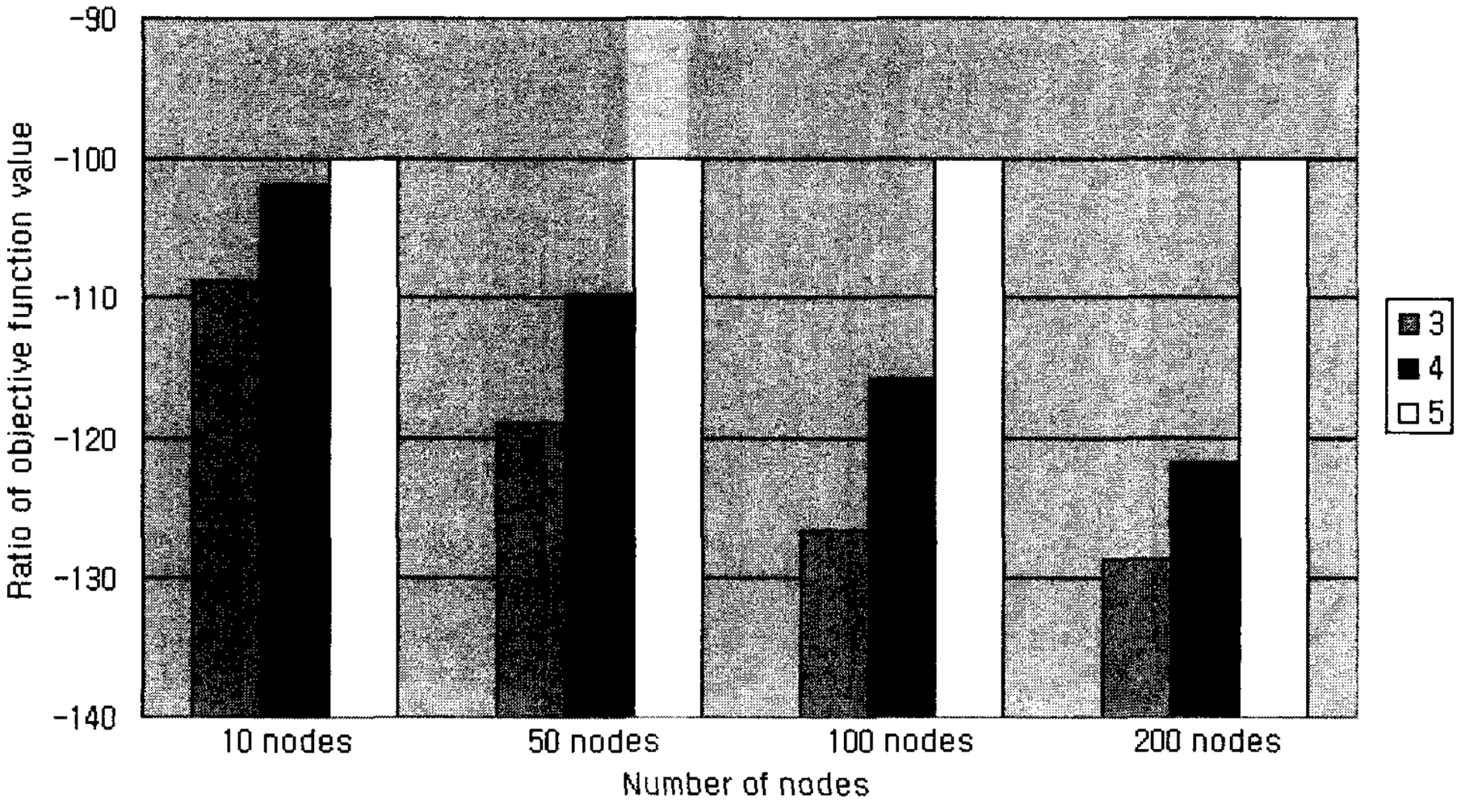


[Figure 6] Test of N_max

<Table 2> shows the result of reliable overlay multicast trees with 10, 50, 100 and 200 nodes. CPLEX [13] is employed to compare the solutions. Due to the exponential growth of branches in the process of CPLEX, it fails to obtain the optimal solution even with a running time of 10,000 seconds for problems with 50, 100 and 200 nodes. The table shows that the proposed tabu search generates optimal solutions in all cases with 10 nodes. For other problems, the solutions by the tabu search are comparable to the bounds obtained by the CPLEX. The average gap from the lower bound is 1.6%, 2.2%, and 3.2% in problems with 50, 100, and 200 nodes.



[Figure 7] Test for D_max



[Figure 8] The effect of node degree

〈Table 2〉 Computational Result of Tabu Search

Problem	10 nodes			50 nodes			100 nodes			200 nodes		
	tabu search	CPLEX	GAP*	tabu search	CPLEX	GAP*	tabu search	CPLEX	GAP*	tabu search	CPLEX	GAP*
1	-2.35 (0.5)	-2.35 (1.6)	0.000	-29.09 (7.1)	-28.89 (10000.0**)	0.007	-80.19 (16.2)	-79.25 (10000.0**)	0.012	-132.85 (41.5)	-130.15 (10000.0**)	0.020
2	-2.52 (0.6)	-2.52 (1.8)	0.000	-31.87 (8.1)	-31.15 (10000.0**)	0.023	-83.98 (19.7)	-83.54 (10000.0**)	0.005	-153.62 (35.6)	-149.35 (10000.0**)	0.028
3	-2.63 (0.8)	-2.63 (1.8)	0.000	-28.68 (7.4)	-28.26 (10000.0**)	0.015	-82.42 (17.8)	-81.23 (10000.0**)	0.014	-139.68 (42.5)	-133.22 (10000.0**)	0.046
4	-2.38 (0.5)	-2.38 (1.6)	0.000	-28.08 (7.2)	-26.95 (10000.0**)	0.040	-81.34 (17.6)	-80.25 (10000.0**)	0.013	-127.35 (30.8)	-121.92 (10000.0**)	0.043
5	-2.42 (0.7)	-2.42 (1.9)	0.000	-28.64 (6.3)	-28.28 (10000.0**)	0.013	-86.23 (19.2)	-84.56 (10000.0**)	0.019	-131.96 (35.9)	-127.64 (10000.0**)	0.033
6	-2.61 (0.7)	-2.61 (1.8)	0.000	-29.37 (6.8)	-29.02 (10000.0**)	0.012	-84.84 (17.5)	-82.48 (10000.0**)	0.028	-146.74 (34.2)	-142.32 (10000.0**)	0.030
7	-2.86 (0.9)	-2.86 (1.4)	0.000	-29.46 (7.2)	-29.16 (10000.0**)	0.010	-87.08 (16.8)	-83.34 (10000.0**)	0.043	-132.69 (36.8)	-128.85 (10000.0**)	0.029
8	-2.48 (0.9)	-2.48 (1.9)	0.000	-30.52 (7.5)	-30.11 (10000.0**)	0.013	-84.76 (18.6)	-81.96 (10000.0**)	0.033	-138.27 (39.2)	-134.28 (10000.0**)	0.029
9	-2.56 (0.6)	-2.56 (1.8)	0.000	-28.65 (7.2)	-28.2 (10000.0**)	0.016	-85.95 (16.4)	-83.88 (10000.0**)	0.024	-126.02 (36.8)	-121.91 (10000.0**)	0.033
10	-2.81 (0.7)	-2.81 (1.7)	0.000	-28.94 (7.5)	-28.55 (10000.0**)	0.013	-88.69 (18.6)	-86.24 (10000.0**)	0.028	-157.61 (35.2)	-152.37 (10000.0**)	0.033

주) 1. * GAP = (tabu search-CPLEX)/tabu search.
2. ** Terminated by the time limit.
3. The numbers in the parenthesis represent the CPU seconds.

Now, we examine the sensitivity of the node degree constraint. [Figure 8] shows the effect of node degree bound. To analyze the effect of node degree, the node degree of all nodes except the source is fixed to 3, 4 and 5 respectively. As shown in the figure, the reliability of overlay multicast tree is very sensitive to the node degree. The improvement of objective function value is vivid in problems with large number of nodes. This is mainly due to the fact that as the node degree increases, the tree has better chance to select links with higher reliability near to the source node. The tree reliability is increased by more than 20% in problems with 100 and 200 nodes when the node degree is increased from three to five. However, as the delay bound (hop count from the source to a member node) becomes tight, the degree constraint becomes more important to have a feasible multicast tree. This clearly leads to lower tree reliability as the degree bound increases.

5. Conclusion

An end-to-end reliable packet delivery problem in overlay networks is considered for next generation multicast service in Internet. Path-level reliabilities from a source to multicast group members are examined to build a reliable overlay multicast tree. The packet processing capability of each member node is considered with degree bound to count the links to other hosts for receiving and forwarding multicast packets. The problem is formulated as a delay bounded minimum longest path which is a well-known NP-complete problem.

A tabu search heuristic is developed based on the swap and reconnection moves. In swap move, a target node with the lowest sojourn probability is swapped with a node having higher probability. In reconnection, a target node is selected and reconnected to a node that satisfies the degree bound with the highest sojourn probability. All children nodes follow the target node in the move. Diversification with long-term memory is also implemented by generating a new solution that connects nodes with higher hop counts closer to the source node.

The performance of the proposed tabu search is experimented with four different sets of overlay networks. Outstanding performance is illustrated by the proposed heuristic. The average gap from the solution by CPLEX is within 3.2% in problems with 100 and 200 nodes. The experiment also shows that the multicast tree reliability is largely dependent on the node degree bound.

참 고 문 헌

[1] Chu Y.-H., Ganjam, A., Ng, TSE, Rao, S.G., Sripanidkulchai, K., Zhan, J., and Zhang, H., Early experience with an Internet broadcast system based on overlay multicast, Technical Report CMU-CS-03-214, Carnegie Mellon University, December 2003.

[2] Diot, C., Levine, B.N., Lyles, B., Kassem, H., and Balensiefen, D., "Deployment issues for the IP multicast service and architecture," *IEEE Network*, Vol.14, No.1(2000), pp.78-88.

[3] Oliveira, C.A.S. and Pardalos, P.M., Algorithms for the streaming cache placement problem on multicast networks, Proceedings of seventh INFORMS telecommunications conference (ITC'04), Florida; Boca Raton : March 7-10, 2004.

[4] Lee, C., Park, H., and Baek, J., An overlay multicast to minimize end-to-end delay in IP networks, Proceedings of tenth international conference on communication technology (ICCT'06), (2006), pp.27-30.

[5] Shi, S. and Turner, J., Routing in overlay multicast networks, IEEE INFOCOM, 2002.

[6] Cho, H. and Lee, C., "Multicast tree rearrangement to recover node failure in overlay multicast network," *Computers and Operations Research*, Vol.33, No.3(2006), pp.581-594.

[7] Lee, C. and Kim, H., "Reliable overlay multicast trees for private internet broadcasting with multiple sessions," *Computers and Operations Research*, Vol.34, No.9(2007), pp.884-899.

[8] Shi, S., Turner, J., and Waldvogel, M., Dimensioning server access bandwidth and multicast routing in overlay networks, In : Proceedings of eleventh international workshop on network and operating systems support for digital audio and video (NOSSDAV'01), (2001), pp.25-26.

[9] Garey, M.R., and Johnson, D.S., Computers and intractability : a guide to the theory of NP-completeness. W.H. Freeman and Company, 1979.

[10] Glover, F. and Laguna, M., Tabu search. Dordrecht : Kluwer Academic Publishers; 1997.

[11] Glover, F., "Tabu search : a tutorial," *Interfaces*, Vol.20, No.4(1990), pp.74-94.

[12] Digital encoding services, Inc., <http://www.digital-encoding.com/streaming_enc.htm>.

[13] CPLEX 8.1, <http://www.ilog.com/products/cplex>.

[14] Common log format, <http://www.w3.org/Daemon/User/Config/Logging.html>.

[15] Extended common log format, <http://www.w3.org/TR/WD-logfile.html>.